Government Smart Card Interoperability Specification v2.1 (NISTIR 6887, 2003 Edition)

Virtual Card Edge Interface Virtual Machine Cards

Conformance Test Instantiation, Verification, and Reporting Scenarios

DRAFT

Alan Goldfine

February 10,2005

This document contains the conformance test instantiation, verification, and reporting scenarios for the APDUs comprising the Virtual Machine Card Virtual Card Edge Interface of version 2.1 of the Government Smart Card Interoperability Specification (GSC-IS), as contained in NIST Interagency Report 6887, 2003 Edition.

The 12 sections of this document correspond to the 12 sections in the Virtual Machine Virtual Card Edge Interface Conformance Test Assertions document.

Appendix A contains the list of constant variables (symbolic constants) used in this document. A constant variable is indicated by a leading underscore (e.g., _goodPIN).

Appendix B contains a description of a card that would be sufficient for the testing of candidate Virtual Machine Virtual Card Edge implementations, when using test suites built according to these scenarios.

APDUs are written with all 7 possible fields present, and with | used as the separator of the non-mandatory fields, i.e., in the form

CLA INS P1 P2|L_c|Data Field|L_e|

where some or all of L_c, Data Field, and L_e may be empty. For example,

```
00 B0 00 06| | |08 represents the APDU
```

00B0000608.

1. SELECT APPLET

Starting State for Each Test:

1. A card that claims to implement the GSC-IS, Version 2.1, is in a reader.

2. The card contains an applet whose AID is _applet1AID. The length of this AID is _applet1AIDLen. This applet contains an object with object ID _object11ID.

3. The card does not contain an applet whose AID is _badAppletAID. The length of this AID is _badAppletAIDLen.

Test for Assertion 1.1

The APDU is tested using valid parameters.

Instantiation Scenario

1. (Pre) Construct the starting state for SELECT APPLET.

2. Issue the following SELECT APPLET APDU:

   00 A4 04 00|_applet1AIDLen|_applet1AID| |

Verification Goal:
To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 90 00 (successful execution)
   - the applet whose AID is _applet1AID is selected.

Perform this verification by attempting to select a container managed by the selected applet.

Verification and Reporting Scenario:

1. **Case 1**: If the SELECT APPLET APDU returns response code == 90 00, then verify that the applet identified by _applet1AID was indeed selected:

   Issue the following SELECT OBJECT APDU:

   00 A4 02 00|02|_object11ID| |

   Case 1.1: If the SELECT OBJECT APDU returns response code == 90 00, then print
   "SELECT APPLET tested using valid parameters has been verified because a subsequent SELECT OBJECT was successful.
   Status: Test 1.1 Passed."

   Case 1.2: If the SELECT OBJECT APDU does not return response code == 90 00, then print
   "SELECT APPLET tested using valid parameters has not been verified because a subsequent SELECT OBJECT was unsuccessful.
   Status: Test 1.1 Failed."

   Case 2: If the SELECT APPLET APDU does not return response code == 90 00, then print

"SELECT APPLET tested using valid parameters has not been verified because the command did not return the correct response code.
Status: Test 1.1 Failed."

Test for Assertion 1.2

The APDU is tested in the case where the applet is logically deleted.

This test is not part of the current version of the GSC-IS Version 2.1 Card Edge Interface Test Suite.

Test for Assertion 1.3

The APDU is tested using an invalid L_c.

This test is not part of the current version of the GSC-IS Version 2.1 Card Edge Interface Test Suite.

Test for Assertion 1.4

The APDU is tested where the specified applet is not on the card.

Instantiation Scenario

1. (Pre) Construct the starting state for SELECT APPLET.

2. Issue the following SELECT APPLET APDU:

   00 A4 04 00|_badAppletAIDLen|_badAppletAID| |

Verification Goal:
To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 69 99 (Applet select failed) or 6A 82 (Applet not found).

Verification and Reporting Scenario:

1. **Case 1**: If the SELECT APPLET APDU returns either of the response codes == 69 99 or 6A 82, then print
   "SELECT APPLET tested where the specified applet is not on the card has been verified.
   Status: Test 1.4 Passed."

   Case 2: If the SELECT APPLET APDU does not return either of the response codes == 69 99 or 6A 82, then print
   "SELECT APPLET tested where the specified applet is not on the card has not been verified because the command did not return the correct response code.
   Status: Test 1.4 Failed."

Test for Assertion 1.5

The APDU is tested using invalid parameters P1-P2.

Instantiation Scenario

1. (Pre) Construct the starting state for SELECT APPLET.

2. Issue the following SELECT APPLET APDU:

```
00 A4 04 FE|_applet1AIDLen|_applet1AID|  |
```

Verification Goal:

To verify the expected results:

1. The APDU returns
    - SW1 SW2 == 6A 86 (Incorrect P1 or P2 parameter).

Verification and Reporting Scenario:

1. **Case 1**: If the SELECT APPLET APDU returns response code == 6A 86, then print

    "SELECT APPLET tested with invalid parameters P1-P2 has been verified.
    Status: Test 1.5 Passed."

    Case 2: If the SELECT APPLET APDU does not return response code == 6A 86, then print

    "SELECT APPLET tested with invalid parameters P1-P2 has not been verified because the command did not return the correct response code.
    Status: Test 1.5 Failed."

2. SELECT OBJECT

Starting State for Each Test:

1. A card that claims to implement the GSC-IS, Version 2.1, is in a reader.

2. The card contains an applet whose AID is _applet1AID. The length of this AID is _applet1AIDLen.

3. The applet whose AID is _applet1AID contains an object with object ID _object11ID. This object contains _existingValue1 in bytes 02 through 05 of its V-buffer. All the services of this object have access control rule ACR_ALWAYS.

4. The applet whose AID is _applet1AID does not contain an object with object ID _badObjectID.

5. Issue the following SELECT APPLET APDU:

   00 A4 04 00|_applet1AIDLen|_applet1AID| |.

Test for Assertion 2.1

The APDU is tested using valid parameters.

Instantiation Scenario

1. (Pre) Construct the starting state for SELECT OBJECT.

2. Issue the following SELECT OBJECT APDU:

   00 A4 02 00|02|_object11ID| |

Verification Goal:
To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 90 00 (successful execution).

2. The object whose ID is _object11ID is selected.

Perform this verification by reading a value in the object.

Verification and Reporting Scenario:

1. **Case 1**: If the SELECT OBJECT APDU returns response code == 90 00, then:

   Issue the following READ BUFFER APDU:

   80 52 00 02|02 02|04| |

   Case 1.1: If the READ BUFFER APDU returns
   - response code == 90 00
   - response data field == _existingValue1

   then print
   "SELECT OBJECT tested using valid parameters has been verified because a subsequent READ BUFFER returned the correct data value.

```
    Status: Test 2.1 Passed."

  Case 1.2: If the READ BUFFER APDU returns
     •  response code == 61 04
  then issue the following GET RESPONSE APDU:
          00 C0 00 00|  |  |04|

    Case 1.2.1: If the GET RESPONSE APDU returns
      •  response code == 90 00
      •  response data field == _existingValue1
    then print
      "SELECT OBJECT tested using valid parameters has been
      verified because a subsequent READ BUFFER returned the
      correct data value.
      Status: Test 2.1 Passed."

    Case 1.2.2: If the GET RESPONSE APDU returns
      •  response code 90 00
      •  response data field /= _existingValue1
    then print
      "SELECT OBJECT tested using valid parameters has not been
      verified because a subsequent READ BUFFER did not return
      the correct data value.
      Status: Test 2.1 Failed."

    Case 1.2.3: If the GET RESPONSE APDU returns
      •  response code /= 90 00
    then print
      "SELECT OBJECT tested using valid parameters has not been
      verified because a subsequent READ BUFFER was ambiguous.
      Status: Test 2.1 Undetermined."

  Case 1.3: If the READ BUFFER APDU returns
     •  response code /= 90 00 or 61 04
  then print
    "SELECT OBJECT tested using valid parameters has not been
    verified because a subsequent READ BUFFER was ambiguous.
    Status: Test 2.1 Undetermined."

Case 2: If the SELECT OBJECT APDU does not return response code
== 90 00, then print
  "SELECT APPLET tested using valid parameters has not been
  verified because the command did not return the correct
  response code.
  Status: Test 2.1 Failed."
```

Test for Assertion 2.2

The APDU is tested using an invalid L_c.

This test is not part of the current version of the GSC-IS Version 2.1 Card Edge Interface Test Suite.

Test for Assertion 2.3

The APDU is tested where the specified object is not managed by the current applet.

Instantiation Scenario

1. (Pre) Construct the starting state for SELECT OBJECT.

2. Issue the following SELECT OBJECT APDU:

```
00 A4 02 00│02│_badObjectID│  │
```

Verification Goal:
To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 6A 82 (file not found).

Verification and Reporting Scenario:

1. **Case 1**: If the SELECT OBJECT APDU returns response code == 6A 82, then print
   "SELECT OBJECT tested where the specified object is not managed by the current applet has been verified.
   Status: Test 2.3 Passed."

   Case 2: If the SELECT OBJECT APDU does not return response code == 6A 82, then print
   "SELECT OBJECT tested where the specified object is not managed by the current applet has not been verified because the command did not return the correct response code.
   Status: Test 2.3 Failed."

Test for Assertion 2.4

The APDU is tested using invalid parameters P1-P2.

Instantiation Scenario

1. (Pre) Construct the starting state for SELECT OBJECT.

2. Issue the following SELECT OBJECT APDU:

```
00 A4 02 FE│02│_object11ID│  │.
```

Verification Goal:
To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 6A 86 (incorrect P1 or P2 parameter).

Verification and Reporting Scenario:

1. **Case 1**: If the SELECT OBJECT APDU returns response code == 6A 86, then print
   "SELECT OBJECT tested using an incorrect P1 or P2 parameter has been verified.
   Status: Test 2.4 Passed."

   Case 2: If the SELECT OBJECT APDU does not return response code == 6A 86, then print

"SELECT OBJECT tested using an incorrect P1 or P2 parameter has not been verified because the command did not return the correct response code.
Status: Test 2.4 Failed."

3. GET PROPERTIES

Starting State for Each Test:

1. A card that claims to implement the GSC-IS, Version 2.1, is in a reader.

2. The card contains an applet whose AID is _applet1AID. The length of this AID is _applet1AIDLen. This applet manages precisely one object, whose object ID is _object11ID. This object
   - contains _existingValue1, of length 4, in bytes 02 through 05 of its V-buffer
   - contains _existingValue2, of length 8, in bytes 06 through 0D of its V-buffer.

3. Issue the following SELECT APPLET APDU:

   00 A4 04 00|_applet1AIDLen|_applet1AID| |.

Test for Assertion 3.1

The APDU is tested using valid parameters (P1 == 01, all properties).

Instantiation Scenario

1. (Pre) Construct the starting state for GET PROPERTIES.

2. Issue the following GET PROPERTIES APDU:

   00 56 01 00| | |19|

Verification Goal:
To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 90 00 (successful execution)
   - the response data field == the following packed TLV buffer:
     01 05 _applet1AID-family _applet1AID-version 40 01 02
     50 0B 41 02 _object11ID 42 05 tag-type 0006 000E.

Verification and Reporting Scenario:

1. **Case 1**: If the GET PROPERTIES APDU returns response code == 90 00, then verify that the response data field contains the properties of the currently selected applet:

   Case 1.1: If
   - bytes 07-11 == 40 01 02 50 0B 41 02 _object11ID 42 05
   - bytes 13-16 == 00 06 00 0E

   then print
   "GET PROPERTIES tested using valid parameters (all properties) has been verified.
   Status: Test 3.1 Passed."

   Case 1.2: If it is not true that
   - bytes 07-11 == 40 01 02 50 0B 41 02 _object11ID 42 05
   - bytes 13-16 == 00 06 00 0E

   then print

"GET PROPERTIES tested using valid parameters (all properties) has not been verified because the APDU did not return the correct properties.
Status: Test 3.1 Failed."

Case 3: If the GET PROPERTIES APDU does not return response code == 90 00, then print

"GET PROPERTIES tested using valid parameters (all properties) has not been verified because the APDU did not return the correct response code.
Status: Test 3.1 Failed."

Test for Assertion 3.2

The APDU is tested using valid parameters (P1 == 02, properties associated with specified tags).

Instantiation Scenario

1. (Pre) Construct the starting state for GET PROPERTIES.

2. Issue the following GET PROPERTIES APDU:

   00 56 02 00|01|50|0D|

Verification Goal:
To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 90 00 (successful execution)
   - the response data field == the following packed TLV buffer:
     50 0B 41 02 _object11ID 42 05 tag-type 0006 000E

   or
   - SW1 SW2 == 61 1A (normal processing, data still available)
   - the available data == the following packed TLV buffer:
     50 0B 41 02 _object11ID 42 05 tag-type 0006 000E.

Verification and Reporting Scenario:

1. **Case 1**: If the GET PROPERTIES APDU returns response code == 90 00, then verify that the response data field contains the specified properties of the currently selected applet:

   Case 1.1: If
   - bytes 00-07 == 50 0B 41 02 _object11ID 42 05
   - bytes 09-0C == 00 06 00 0E

   then print

   "GET PROPERTIES tested using valid parameters (properties associated with specified tags) has been verified.
   Status: Test 3.2 Passed."

   Case 1.2: If it is not true that
   - bytes 00-07 == 50 0B 41 02 _object11ID 42 05
   - bytes 09-0C == 00 06 00 0E

   then print

   "GET PROPERTIES tested using valid parameters (properties associated with specified tags) has not been verified because the APDU did not return the correct properties.
   Status: Test 3.2 Failed."

Case 2: If the GET PROPERTIES APDU returns response code == 61 0D, then issue the following GET RESPONSE:

```
00 C0 00 00|  |  |0D|
```

Case 2.1: If the GET RESPONSE returns response code 90 00, then verify that the response data field contains the specified properties of the specified object:

Case 2.1.1: If

- bytes 00-07 == 50 0B 41 02 _object11ID 42 05
- bytes 09-0C == 00 06 00 0E

then print

"GET PROPERTIES tested using valid parameters (properties associated with specified tags) has been verified.
Status: Test 3.2 Passed."

Case 2.1.2: If it is not true that

- bytes 00-07 == 50 0B 41 02 _object11ID 42 05
- bytes 09-0C == 00 06 00 0E

then print

"GET PROPERTIES tested using valid parameters (properties associated with specified tags) has not been verified because a subsequent GET RESPONSE did not return the correct properties.
Status: Test 3.2 Failed."

Case 2.2: If the GET RESPONSE does not return response code 90 00, then print

"GET PROPERTIES tested using valid parameters (properties associated with specified tags) has not been verified because a subsequent GET RESPONSE did not return the correct response code.
Status: Test 3.2 Failed."

Case 3: If the GET PROPERTIES APDU does not return response code == 90 00 or 61 0D, then print

"GET PROPERTIES tested using valid parameters (properties associated with specified tags) has not been verified because the APDU did not return the correct response code.
Status: Test 3.2 Failed."

Test for Assertion 3.3

The APDU is tested using an incorrect parameter L_c.

This test is not part of the current version of the GSC-IS Version 2.1 Card Edge Interface Test Suite.

Test for Assertion 3.4

The APDU is tested using an incorrect parameter L_e (P1 == 01).

Instantiation Scenario

1. (Pre) Construct the starting state for GET PROPERTIES.

2. Issue the following GET PROPERTIES APDU:

   00 56 01 00| | |FE|

Verification Goal:

To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 6C 1D (incorrect parameter L_e)

Verification and Reporting Scenario:

1. **Case 1**: If the GET PROPERTIES APDU returns response code == 6C 1D, then print

   "GET PROPERTIES tested using an incorrect parameter L_e has been verified.
   Status: Test 3.4 Passed."

   Case 2: If the GET PROPERTIES APDU does not return response code == 6C 1D, then print

   "GET PROPERTIES tested using an incorrect parameter L_c has not been verified because the command did not return the correct response code.
   Status: Test 3.4 Failed."

Test for Assertion 3.5

The APDU is tested using invalid parameters in the command data field (DATA = FE).

Instantiation Scenario

1. (Pre) Construct the starting state for GET PROPERTIES.

2. Issue the following GET PROPERTIES APDU:

   00 56 02 00|01|FE|1A|

Verification Goal:

To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 6A 80 (invalid parameters in the command data field)

Verification and Reporting Scenario:

1. **Case 1**: If the GET PROPERTIES APDU returns response code == 6A 80, then print

   "GET PROPERTIES tested using invalid parameters in the command data field has been verified.
   Status: Test 3.5 Passed."

   Case 2: If the GET PROPERTIES APDU does not return response code == 6A 80, then print

   "GET PROPERTIES tested using invalid parameters in the command data field has not been verified because the command did not return the correct response code.
   Status: Test 3.5 Failed."

Test for Assertion 3.6

The APDU is tested using invalid parameters P1-P2.

Instantiation Scenario

1. (Pre) Construct the starting state for GET PROPERTIES.

2. Issue the following GET PROPERTIES APDU:

```
00 56 FE 00|  |  |1A|
```

Verification Goal:

To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 6A 86 (incorrect P1 or P2 parameter)

Verification and Reporting Scenario:

1. **Case 1**: If the GET PROPERTIES APDU returns response code == 6A 86, then print

   "GET PROPERTIES tested using an invalid P1 or P2 parameter has been verified.
   Status: Test 3.6 Passed."

   Case 2: If the GET PROPERTIES APDU does not return response code == 6A 86, then print

   "GET PROPERTIES tested using an invalid P1 or P2 parameter has not been verified because the command did not return the correct response code.
   Status: Test 3.6 Failed."

4. GET ACR

Starting State for Each Test:

1. A card that claims to implement the GSC-IS, Version 2.1, is in a reader.

2. The card contains an applet whose AID is _applet2AID. This applet manages precisely one object whose object ID is _object21ID. All the services of this object have ACR type ACR_XAUTH_AND_PIN.

3. The card contains a second applet whose AID is _applet3AID. This applet manages precisely one object, whose object ID is _object31ID. All the services of this object have ACR type ACR_PIN.

4. There are two ACR entries in the ACR table in the card's CCC:
   - ACRID == _ACR1ID, corresponding to ACR_PIN
   - ACRID == _ACR2ID, corresponding to ACR_XAUTH_AND_PIN.

5. On the card, XAUTH is managed by an access method provider applet with
   - AID == _appletXAUTHAID
   - applet ID == _appletXAUTHID
   - keyIDOrReference == 01.

   PIN is managed by an access method provider applet with
   - AID == _appletPINAID
   - applet ID == _appletPINID
   - keyIDOrReference == 01.

6. Issue the following SELECT APPLET APDU:

   ```
   00 A4 04 00|_applet3AIDLen|_applet3AID|  |.
   ```

(Need to coordinate above card description)

Test for Assertion 4.1

The APDU is tested using valid parameters (P1 == 00, complete ACR table).

Instantiation Scenario

1. (Pre) Construct the starting state for GET ACR.

2. Issue the following GET ACR APDU:

   ```
   80 4C 00 00|  |  |  |
   ```

Verification Goal:

To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 90 00 (successful execution)
   - the response data field == the card's Applet Information String followed by the card's complete ACR Table.

Verification and Reporting Scenario:

1. Let
   - R0 == the bytes A1 01 02
   - R1 == the bytes A0 05 _ACR1ID 06 01 _appletPINID 01
   - R2 == the bytes A0 07 _ACR2ID 07 02
   - R3 == the bytes _appletPINID 01
   - R4 == the bytes _appletXAUTHID 01

   Case 1: If the GET ACR APDU returns response code == 90 00, then verify that the response data field contains the card's complete ACR Table:

Note: The Applet Information String is implementor dependent, and so is not examined.

Case 1.1: If bytes 07-19 of the response data field are

- R0 R1 R2 R3 R4

or

- R0 R1 R2 R4 R3

or

- R0 R2 R3 R4 R1

or

- R0 R2 R4 R3 R1

then print

"GET ACR tested using valid parameters (complete ACR table) has been verified.
Status: Test 4.1 Passed."

Case 1.2: If it is not true that bytes 07-19 of the response data field are

- R0 R1 R2 R3 R4

or

- R0 R1 R2 R4 R3

or

- R0 R2 R3 R4 R1

or

- R0 R2 R4 R3 R1

then print

"GET ACR tested using valid parameters (complete ACR table) has not been verified because the APDU did not return the correct table.
Status: Test 4.1 Failed."

Case 2: If the GET ACR APDU does not return response code == 90 00, then print

"GET ACR tested using valid parameters (complete ACR table) has not been verified because the APDU did not return the correct response code.
Status: Test 4.1 Failed."

Test for Assertion 4.2

The APDU is tested using valid parameters (P1 == 01, single ACR table entry).

Instantiation Scenario

1. (Pre) Construct the starting state for GET ACR.

2. Issue the following GET ACR APDU:

```
80 4C 01 00|01|_ACR2ID|  |
```

Verification Goal:

To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 90 00 (successful execution)
   - the response data field == the card's Applet Information String followed by the card's ACR Table entry for _ACR2ID.

   or

   - SW1 SW2 == 61 09 (normal processing, data still available)
   - the available data == the card's Applet Information String followed by the card's ACR Table entry for _ACR2ID.

Verification and Reporting Scenario:

1. Let
   - R0 == the bytes A0 07 _ACR2ID 07 02
   - R1 == the bytes _appletPINID 01
   - R2 == the bytes _appletXAUTHID 01

Note: The Applet Information String is implementor dependent, and so is not examined.

Case 1: If the GET ACR APDU returns response code == 90 00, then verify that the response data field contains the card's ACR Table entry for _ACR2ID:

Case 1.1: If bytes 07-0F of the response data field are

- R0 R1 R2

or

- R0 R2 R1

then print

"GET ACR tested using valid parameters (single ACR table entry) has been verified.
Status: Test 4.2 Passed."

Case 1.2: If it is not true that bytes 07-0F of the response data field are

- R0 R1 R2

or

- R0 R2 R1

then print

"GET ACR tested using valid parameters (single ACR table entry) has not been verified because the APDU did not return the correct table entry.
Status: Test 4.2 Failed."

Case 2: If the GET ACR APDU returns response code == 61 09, then issue the following GET RESPONSE:

```
00 C0 00 00|  |  |09|
```

Case 2.1: If the GET RESPONSE returns response code 90 00, then verify that the response data field contains the card's ACR Table entry for _ACR2ID:

Case 2.1.1: If bytes 07-0F of the response data field are

- R0 R1 R2

or

- R0 R2 R1

then print

"GET ACR tested using valid parameters (single ACR table entry) has been verified.
Status: Test 4.2 Passed."

Case 2.1.2: If it is not true that bytes 07-0F of the response data field are

- R0 R1 R2

or

- R0 R2 R1

then print

"GET ACR tested using valid parameters (single ACR table entry) has not been verified because the APDU did not return the correct table entry.
Status: Test 4.2 Failed."

Case 2.2: If the GET RESPONSE does not return response code 90 00, then print

"GET ACR tested using valid parameters (single ACR table entry) has not been verified because a subsequent GET RESPONSE did not return the correct response code.
Status: Test 4.2 Failed."

Case 3: If the GET ACR APDU does not return response code == 90 00 or 61 10, then print

"GET ACR tested using valid parameters (single ACR table entry) has not been verified because the APDU did not return the correct response code.
Status: Test 4.2 Failed."

Test for Assertion 4.3

The APDU is tested using valid parameters (P1 == 10, complete Applet/Object ACR table).

This test is not part of the current version of the GSC-IS Version 2.1 Card Edge Interface Test Suite.

Test for Assertion 4.4

The APDU is tested using valid parameters (P1 == 11, single Applet/Object ACR table entry for one applet based on applet ID).

This test is not part of the current version of the GSC-IS Version 2.1 Card Edge Interface Test Suite.

Test for Assertion 4.5

The APDU is tested using valid parameters (P1 == 12, single Applet/Object ACR table entry for one object based on object ID).

This test is not part of the current version of the GSC-IS Version 2.1 Card Edge Interface Test Suite.

Test for Assertion 4.6

The APDU is tested using valid parameters (P1 == 20, complete Access Method Provider table).

Instantiation Scenario

1. (Pre) Construct the starting state for GET ACR.

2. Issue the following GET ACR APDU:

   80 4C 20 00| | | |

Verification Goal:

To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 90 00 (successful execution)
   - the response data field == the card's Applet Information String followed by the card's complete Access Method Provider Table.

Verification and Reporting Scenario:

1. Let
   - R0 == the bytes 91 01 02
   - R1 == the bytes 90 0A _appletPINID 92 07 _appletPINAID
   - R2 == the bytes 90 0A _appletXAUTHID 92 07 _appletXAUTHAID

Case 1: If the GET ACR APDU returns response code == 90 00, then verify that the response data field contains the card's complete Access Method Provider Table:

Case 1.1: If bytes 00-1A of the response data field are

- R0 R1 R2

or

- R0 R2 R1

then print

"GET ACR tested using valid parameters (complete Access Method Provider table) has been verified.
Status: Test 4.6 Passed."

Case 1.2: If it is not true that bytes 00-1A of the response data field are

- R0 R1 R2

or

- R0 R2 R1

then print

"GET ACR tested using valid parameters (complete Access Method Provider table) has not been verified because the APDU did not return the correct table.

Status: Test 4.6 Failed."

Case 2: If the GET ACR APDU does not return response code == 90 00, then print
"GET ACR tested using valid parameters (complete Access Method Provider table) has not been verified because the APDU did not return the correct response code.
Status: Test 4.6 Failed."

Test for Assertion 4.7

The APDU is tested using valid parameters (P1 == 21, complete Service Applet table).

This test is not part of the current version of the GSC-IS Version 2.1 Card Edge Interface Test Suite.

Test for Assertion 4.8

The APDU is tested in the case where the specified applet is logically deleted.

This test is not part of the current version of the GSC-IS Version 2.1 Card Edge Interface Test Suite.

Test for Assertion 4.9

The APDU is tested using an incorrect parameter L_c.

This test is not part of the current version of the GSC-IS Version 2.1 Card Edge Interface Test Suite.

Test for Assertion 4.10

The APDU is tested in the case where the specified applet is not on the card (Applet/Object ACR table entries for one applet based on applet AID).

This test is not part of the current version of the GSC-IS Version 2.1 Card Edge Interface Test Suite.

Test for Assertion 4.11

The APDU is tested using an incorrect P1 or P2 parameter.

Instantiation Scenario

1. (Pre) Construct the starting state for GET ACR.

2. Issue the following GET ACR APDU:

```
        80 4C 00 FE|  |  |  |
```

Verification Goal:

To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 6A 86 (incorrect P1 or P2 parameter)

Verification and Reporting Scenario:

1. **Case 1**: If the GET ACR APDU returns response code == 6A 86, then print

   "GET ACR tested using an incorrect P1 or P2 parameter has been verified.
   Status: Test 4.11 Passed."

   Case 2: If the GET ACR APDU does not return response code == 6A 86, then print

   "GET ACR tested using an incorrect P1 or P2 parameter has not been verified because the command did not return the correct response code.
   Status: Test 4.11 Failed."

5. GET RESPONSE

Starting State for Each Test:

1. A card that claims to implement the GSC-IS, Version 2.1, is in a reader.

2. The card contains an applet whose AID is _applet1AID. This AID has length _applet1AIDLen.

3. The applet whose AID is _applet1AID contains an object with object ID _object11ID. This object
   - contains _tag1 in byte 02, 04 in byte 03, _tag2 in byte 04, and 08 in byte 05 of its T-buffer
   - contains _existingValue1 in bytes 02 through 05, and _existingValue2 in bytes 06-0D of its V-buffer
   - has access control rule ACR_ALWAYS for all services.

4. Select the applet whose AID is _applet1AID:

   Issue the following SELECT APPLET APDU:

   00 A4 04 00|_applet1AIDLen|_applet1AID| |

   Case 1: If the SELECT APPLET returns SW1 SW2 == 90 00, then continue with Step 5.

   Case 2: If the SELECT APPLET does not return SW1 SW2 == 90 00, then print
   "The applet cannot be selected. GET RESPONSE cannot be tested."
   End current test.

5. Select the object whose object ID is _object11ID:

   Issue the following SELECT OBJECT APDU:

   00 A4 02 00|02|_object11ID| |

   Case 1: If the SELECT OBJECT returns SW1 SW2 == 90 00, then continue with Step 6.

   Case 2: If the SELECT OBJECT does not return SW1 SW2 == 90 00, then print
   "The object cannot be selected. GET RESPONSE cannot be tested."
   End current test.

6. Issue the following READ BUFFER APDU:

   80 52 00 06|02|02 04| |

   Case 1: If the READ BUFFER APDU returns SW1 SW2 == 90 00, then print
   "T=1 communications protocol. There is no test for GET RESPONSE in the current version of the GSC-IS Version 2.1 Card Edge Interface Test Suite."
   End current test.

Case 2: If the READ BUFFER APDU returns SW1 SW2 == 61 04, then continue with step 2 of the Instantiation Scenario.

Case 3: If the READ BUFFER APDU does not return either SW1 SW2 == 90 00 or SW1 SW2 == 61 08, then print
"The specified buffer cannot be read. GET RESPONSE cannot be tested."
End current test.

Test for Assertion 5.1

The APDU is tested using valid parameters, with the number of bytes specified to be retrieved equal to the maximum available.

Instantiation Scenario

1. (Pre) Construct the starting state for GET RESPONSE.

2. Issue the following GET RESPONSE APDU:

   00 C0 00 00| | |04|

Verification Goal:
To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 90 00 (successful execution)
   - the response data field == _existingValue2.

Verification and Reporting Scenario:

1. **Case 1**: If GET RESPONSE returns response code == 90 00, then verify that the response data field contains the correct data value:

   Case 1.1: If
   - the response data field == _existingValue2

   then print
   "GET RESPONSE tested using valid parameters has been verified.
   Status: Test 5.1 Passed."

   Case 1.2: If it is not true that
   - the response data field == _existingValue2

   then print
   "GET RESPONSE tested using valid parameters has not been verified because the APDU did not return the correct data value.
   Status: Test 5.1 Failed."

Case 2: If the GET RESPONSE APDU does not return response code == 90 00, then print
"GET RESPONSE tested using valid parameters has not been verified because the APDU did not return the correct response code.
Status: Test 5.1 Failed."

Test for Assertion 5.2

The APDU is tested with the number of bytes specified to be retrieved less than the maximum available.

Instantiation Scenario

1. (Pre) Construct the starting state for GET RESPONSE.

2. Issue the following GET RESPONSE APDU:

   ```
   00 C0 00 00|  |  |02|
   ```

Verification Goal:
To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 61 02 (normal processing, data still available).

Verification and Reporting Scenario:

1. **Case 1**: If GET RESPONSE returns response code == 61 02, then print
   "GET RESPONSE, tested with the number of bytes specified to be retrieved less than the maximum available, has been verified.
   Status: Test 5.2 Passed."

   Case 2: If the GET RESPONSE APDU does not return response code == 61 04, then print
   "GET RESPONSE, tested with the number of bytes specified to be retrieved less than the maximum available, has not been verified because the APDU did not return the correct response code.
   Status: Test 5.2 Failed."

Test for Assertion 5.3

The APDU is tested with the number of bytes specified to be retrieved greater than the maximum available.

Instantiation Scenario

1. (Pre) Construct the starting state for GET RESPONSE.

2. Issue the following GET RESPONSE APDU:

   ```
   00 C0 00 00|  |  |20|
   ```

Verification Goal:
To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 6C 08 (incorrect parameter L_e).

Verification and Reporting Scenario:

1. **Case 1**: If GET RESPONSE returns response code == 6C 08, then print
   "GET RESPONSE, tested with the number of bytes specified to be retrieved greater than the maximum available, has been verified.
   Status: Test 5.3 Passed."

Case 2: If the GET RESPONSE APDU does not return response code == 6C 08, then print

"GET RESPONSE, tested with the number of bytes specified to be retrieved greater than the maximum available, has not been verified because the APDU did not return the correct response code.
Status: Test 5.3 Failed."

Test for Assertion 5.4

The APDU is tested with an invalid P1 or P2 parameter.

Instantiation Scenario

1. (Pre) Construct the starting state for GET RESPONSE.

2. Issue the following GET RESPONSE APDU:

```
00 C0 00 FE|  |  |08|
```

Verification Goal:
To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 6A 86 (invalid P1 or P2 parameter).

Verification and Reporting Scenario:

1. **Case 1**: If GET RESPONSE returns response code == 6A 86, then print

   "GET RESPONSE, tested with an invalid P1 or P2 parameter, has been verified.
   Status: Test 5.4 Passed."

   Case 2: If the GET RESPONSE APDU does not return response code == 6A 86, then print

   "GET RESPONSE, tested with an invalid P1 or P2 parameter, has not been verified because the APDU did not return the correct response code.
   Status: Test 5.4 Failed."

6. VERIFY PIN

Starting State for Each Test:

1. A card that claims to implement the GSC-IS, Version 2.1, is in a reader.

2. The card contains an applet whose AID is _applet1AID. This AID has length _applet1AIDLen.

3. The applet whose AID is _applet1AID contains an object with object ID _object11ID. This object
   - contains _tag1 in byte 02, 04 in byte 03, _tag2 in byte 04, and 08 in byte 05 of its T-buffer
   - contains _existingValue1 in bytes 02 through 05, and _existingValue2 in bytes 06-0D of its V-buffer
   - has access control rule ACR_ALWAYS for all services.

4. The card contains a second applet whose AID is _applet3AID. This AID has length _applet3AIDLen.

5. The applet whose AID is _applet3AID contains an object with object ID _object31ID. This object
   - contains _tag9 in byte 02 and 04 in byte 03 of its T-buffer
   - contains _existingValue9 in bytes 02 through 05 of its V-buffer
   - has access control rule ACR_PIN for all services. _goodPIN is the PIN that has been set for the services of the object, and 3 is the maximum number of PIN tries allowed.

Test for Assertion 6.1

The APDU is tested using valid parameters (PIN verification required, PIN not yet verified, no PIN code specified).

Instantiation Scenario

1. (Pre) Construct the starting state for VERIFY PIN.

2. (Pre) Select the applet whose AID is _applet3AID:

   Issue the following SELECT APPLET APDU:

   00 A4 04 00│_applet3AIDLen│_applet3AID│ │

   Case 1: If the SELECT APPLET returns SW1 SW2 == 90 00, then continue with Step 3.

   Case 2: If the SELECT APPLET does not return SW1 SW2 == 90 00, then print
   "The applet cannot be selected. VERIFY PIN (PIN verification required, PIN not yet verified, no PIN code specified) cannot be tested."
   End current test.

3. (Pre) Select the object whose object ID is _object31ID:

Issue the following SELECT OBJECT APDU:

```
00 A4 02 00|02|_object31ID| |
```

Case 1: If the SELECT OBJECT returns SW1 SW2 == 90 00, then continue with Step 4.

Case 2: If the SELECT OBJECT does not return SW1 SW2 == 90 00, then print
"The object cannot be selected. VERIFY PIN (PIN verification required, PIN not yet verified, no PIN code specified) cannot be tested."
End current test.

4. Issue the following VERIFY PIN APDU:

```
00 20 00 00|00| | |
```

Verification Goal:
To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 63 C2 (PIN not verified).

2. The services of the currently selected object cannot be performed.

Perform this verification by attempting to read a value in the currently selected object.

Verification and Reporting Scenario:

1. **Case 1**: If the VERIFY PIN APDU returns response code == 63 C2, then:

   Issue the following READ BUFFER APDU:

   ```
   80 52 00 02|02|02 04| |
   ```

   Case 1.1: If the READ BUFFER APDU returns response code == 69 82, then print
   "VERIFY PIN tested using valid parameters (PIN verification required, PIN not yet verified, no PIN code specified) has been verified because a subsequent READ BUFFER was not allowed to read the currently selected object.
   Status: Test 6.1 Passed."

   Case 1.2: If the READ BUFFER APDU does not return response code == 69 82, then print
   "VERIFY PIN tested using valid parameters (PIN verification required, PIN not yet verified, no PIN code specified) has not been verified because a subsequent READ BUFFER returned an incorrect return code.
   Status: Test 6.1 Passed."

Case 2: If the VERIFY PIN APDU does not return response code == 63 C2, then print
"VERIFY PIN tested using valid parameters (PIN verification required, PIN not yet verified, no PIN code specified) has not been verified because the command did not return the correct response code.

Status: Test 6.1 Failed."

Test for Assertion 6.2

The APDU is tested using valid parameters (PIN verification required, PIN not yet verified, correct PIN code specified).

Instantiation Scenario

1. (Pre) Construct the starting state for VERIFY PIN.

2. (Pre) Select the applet whose AID is _applet3AID:

   Issue the following SELECT APPLET APDU:

   00 A4 04 00|_applet3AIDLen|_applet3AID| |

   Case 1: If the SELECT APPLET returns SW1 SW2 == 90 00, then continue with Step 3.

   Case 2: If the SELECT APPLET does not return SW1 SW2 == 90 00, then print
   "The applet cannot be selected. VERIFY PIN (PIN verification required, PIN not yet verified, correct PIN code specified) cannot be tested."
   End current test.

3. (Pre) Select the object whose object ID is _object31ID:

   Issue the following SELECT OBJECT APDU:

   00 A4 02 00|02|_object31ID| |

   Case 1: If the SELECT OBJECT returns SW1 SW2 == 90 00, then continue with Step 4.

   Case 2: If the SELECT OBJECT does not return SW1 SW2 == 90 00, then print
   "The object cannot be selected. VERIFY PIN (PIN verification required, PIN not yet verified, correct PIN code specified) cannot be tested."
   End current test.

4. Issue the following VERIFY PIN APDU:

   00 20 00 00|08|_goodPIN| |

Verification Goal:
To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 90 00 (successful execution).

2. The services of the currently selected object can be performed.

Perform this verification by attempting to read a value in the currently selected object.

Verification and Reporting Scenario:

1. **Case 1**: If the VERIFY PIN APDU returns response code == 90 00, then:

   Issue the following READ BUFFER APDU:

   ```
   80 52 00 02|02|02 04| |
   ```

   Case 1.1: If the READ BUFFER APDU returns response code == 90 00 or 61 04, then print
   "VERIFY PIN tested using valid parameters (PIN verification required, PIN not yet verified, correct PIN code specified) has been verified because a subsequent READ BUFFER was allowed to read the currently selected object.
   Status: Test 6.2 Passed."

   Case 1.2: If the READ BUFFER APDU does not return response code == 90 00 or 61 04, then print
   "VERIFY PIN tested using valid parameters (PIN verification required, PIN not yet verified, correct PIN code specified) has not been verified because a subsequent READ BUFFER was not allowed to read the currently selected object.
   Status: Test 6.2 Failed."

   Case 2: If the VERIFY PIN APDU does not return response code == 90 00, then print
   "VERIFY PIN tested using valid parameters (PIN verification required, PIN not yet verified, correct PIN code specified) has not been verified because the command did not return the correct response code.
   Status: Test 6.2 Failed."

Test for Assertion 6.3

The APDU is tested using valid parameters (PIN verification required, PIN not yet verified, incorrect PIN code specified).

Instantiation Scenario

1. (Pre) Construct the starting state for VERIFY PIN.

2. (Pre) Select the applet whose AID is _applet3AID:

   Issue the following SELECT APPLET APDU:

   ```
   00 A4 04 00|_applet3AIDLen|_applet3AID| |
   ```

   Case 1: If the SELECT APPLET returns SW1 SW2 == 90 00, then continue with Step 3.

   Case 2: If the SELECT APPLET does not return SW1 SW2 == 90 00, then print
   "The applet cannot be selected. VERIFY PIN (PIN verification required, PIN not yet verified, incorrect PIN code specified) cannot be tested."
   End current test.

3. (Pre) Select the object whose object ID is _object31ID:

Issue the following SELECT OBJECT APDU:

```
00 A4 02 00|02|_object31ID|  |
```

Case 1: If the SELECT OBJECT returns SW1 SW2 == 90 00, then continue with Step 4.

Case 2: If the SELECT OBJECT does not return SW1 SW2 == 90 00, then print
"The object cannot be selected. VERIFY PIN (PIN verification required, PIN not yet verified, incorrect PIN code specified) cannot be tested."
End current test.

4. Issue the following VERIFY PIN APDU:

```
00 20 00 0008|_badPIN|  |  |
```

Verification Goal:
To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 63 C2 (PIN not verified).

2. The services of the currently selected object cannot be performed.

Perform this verification by attempting to read a value in the currently selected object.

Verification and Reporting Scenario:

1. **Case 1**: If the VERIFY PIN APDU returns response code == 63 C2, then:

   Issue the following READ BUFFER APDU:

   ```
   80 52 00 02|02|02 04|  |
   ```

   Case 1.1: If the READ BUFFER APDU returns response code == 69 82, then print
   "VERIFY PIN tested using valid parameters (PIN verification required, PIN not yet verified, incorrect PIN code specified) has been verified because a subsequent READ BUFFER was not allowed to read the currently selected object.
   Status: Test 6.3 Passed."

   Case 1.2: If the READ BUFFER APDU does not return response code == 69 82, then print
   "VERIFY PIN tested using valid parameters (PIN verification required, PIN not yet verified, incorrect PIN code specified) has not been verified because a subsequent READ BUFFER returned an incorrect return code.
   Status: Test 6.3 Passed."

Case 2: If the VERIFY PIN APDU does not return response code == 63 C2, then print
"VERIFY PIN tested using valid parameters (PIN verification required, PIN not yet verified, incorrect PIN code specified) has not been verified because the command did not return the correct response code.

Status: Test 6.3 Failed."

Test for Assertion 6.4

The APDU is tested using valid parameters (PIN verification required, PIN has already been verified, no PIN code specified).

Instantiation Scenario

1. (Pre) Construct the starting state for VERIFY PIN.

2. (Pre) Select the applet whose AID is _applet3AID:

   Issue the following SELECT APPLET APDU:

   ```
   00 A4 04 00|_applet3AIDLen|_applet3AID|  |
   ```

   Case 1: If the SELECT APPLET returns SW1 SW2 == 90 00, then continue with Step 3.

   Case 2: If the SELECT APPLET does not return SW1 SW2 == 90 00, then print
   "The applet cannot be selected. VERIFY PIN (PIN verification required, PIN has already been verified, no PIN code specified) cannot be tested."
   End current test.

3. (Pre) Select the object whose object ID is _object31ID:

   Issue the following SELECT OBJECT APDU:

   ```
   00 A4 02 00|02|_object31ID|  |
   ```

   Case 1: If the SELECT OBJECT returns SW1 SW2 == 90 00, then continue with Step 4.

   Case 2: If the SELECT OBJECT does not return SW1 SW2 == 90 00, then print
   "The object cannot be selected. VERIFY PIN (PIN verification required, PIN has already been verified, no PIN code specified) cannot be tested."
   End current test.

4. (Pre) Issue the following VERIFY PIN APDU:

   ```
   00 20 00 00|08|_goodPIN|  |
   ```

   Case 1: If the VERIFY PIN returns SW1 SW2 == 90 00, then continue with Step 5.

   Case 2: If the VERIFY PIN does not return SW1 SW2 == 90 00, then print
   "The preliminary VERIFY PIN was unsuccessful. VERIFY PIN (PIN verification required, PIN has already been verified, no PIN code specified) cannot be tested."
   End current test.

5. Issue the following VERIFY PIN APDU:

00 20 00 00|00| | |

Verification Goal:
To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 90 00 (successful execution).

2. The services of the currently selected object can be performed.

Perform this verification by attempting to read a value in the currently selected object.

Verification and Reporting Scenario:

1. **Case 1**: If the VERIFY PIN APDU returns response code == 90 00, then:

   Issue the following READ BUFFER APDU:

   80 52 00 02|02|02 04| |

   Case 1.1: If the READ BUFFER APDU returns response code == 90 00 or 61 04, then print
   "VERIFY PIN tested using valid parameters (PIN verification required, PIN has already been verified, no PIN code specified) has been verified because a subsequent READ BUFFER was allowed to read the currently selected object.
   Status: Test 6.4 Passed."

   Case 1.2: If the READ BUFFER APDU does not return response code == 90 00 or 61 04, then print
   "VERIFY PIN tested using valid parameters (PIN verification required, PIN has already been verified, no PIN code specified) has not been verified because a subsequent READ BUFFER was not allowed to read the currently selected object.
   Status: Test 6.4 Failed."

   Case 2: If the VERIFY PIN APDU does not return response code == 90 00, then print
   "VERIFY PIN tested using valid parameters (PIN verification required, PIN has already been verified, no PIN code specified) has not been verified because the command did not return the correct response code.
   Status: Test 6.4 Failed."

Test for Assertion 6.5

The APDU is tested using valid parameters (PIN verification required, PIN has already been verified, correct PIN code specified).

Instantiatio Scenario

1. (Pre) Construct the starting state for VERIFY PIN.

2. (Pre) Select the applet whose AID is _applet3AID:

   Issue the following SELECT APPLET APDU:

```
00 A4 04 00|_applet3AIDLen|_applet3AID|  |
```

Case 1: If the SELECT APPLET returns SW1 SW2 == 90 00, then continue with Step 3.

Case 2: If the SELECT APPLET does not return SW1 SW2 == 90 00, then print
"The applet cannot be selected. VERIFY PIN (PIN verification required, PIN has already been verified, correct PIN code specified) cannot be tested."
End current test.

3. (Pre) Select the object whose object ID is _object31ID:

   Issue the following SELECT OBJECT APDU:

   ```
   00 A4 02 00|02|_object31ID|  |
   ```

   Case 1: If the SELECT OBJECT returns SW1 SW2 == 90 00, then continue with Step 4.

   Case 2: If the SELECT OBJECT does not return SW1 SW2 == 90 00, then print
   "The object cannot be selected. VERIFY PIN (PIN verification required, PIN has already been verified, correct PIN code specified) cannot be tested."
   End current test.

4. (Pre) Issue the following VERIFY PIN APDU:

   ```
   00 20 00 00|08|_goodPIN|  |
   ```

   Case 1: If the VERIFY PIN returns SW1 SW2 == 90 00, then continue with Step 5.

   Case 2: If the VERIFY PIN does not return SW1 SW2 == 90 00, then print
   "The preliminary VERIFY PIN was unsuccessful. VERIFY PIN (PIN verification required, PIN has already been verified, correct PIN code specified) cannot be tested."
   End current test.

5. Issue the following VERIFY PIN APDU:

   ```
   00 20 00 00|08|_goodPIN|  |
   ```

<u>Verification Goal</u>:
To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 90 00 (successful execution).

2. The services of the currently selected object can be performed.

Perform this verification by attempting to read a value in the currently selected object.

<u>Verification and Reporting Scenario</u>:

1. **Case 1**: If the VERIFY PIN APDU returns response code == 90 00, then:

   Issue the following READ BUFFER APDU:

   ```
   80 52 00 02|02|02 04|  |
   ```

   Case 1.1: If the READ BUFFER APDU returns response code == 90 00 or 61 04, then print
   "VERIFY PIN tested using valid parameters (PIN verification required, PIN has already been verified, correct PIN code specified) has been verified because a subsequent READ BUFFER was allowed to read the currently selected object.
   Status: Test 6.5 Passed."

   Case 1.2: If the READ BUFFER APDU does not return response code == 90 00 or 61 04, then print
   "VERIFY PIN tested using valid parameters (PIN verification required, PIN has already been verified, correct PIN code specified) has not been verified because a subsequent READ BUFFER was not allowed to read the currently selected object.
   Status: Test 6.5 Failed."

   Case 2: If the VERIFY PIN APDU does not return response code == 90 00, then print
   "VERIFY PIN tested using valid parameters (PIN verification required, PIN has already been verified, correct PIN code specified) has not been verified because the command did not return the correct response code.
   Status: Test 6.5 Failed."

Test for Assertion 6.6

The APDU is tested using valid parameters (PIN verification required, PIN has already been verified, incorrect PIN code specified).

Instantiation Scenario

1. (Pre) Construct the starting state for VERIFY PIN.

2. (Pre) Select the applet whose AID is _applet3AID:

   Issue the following SELECT APPLET APDU:

   ```
   00 A4 04 00|_applet3AIDLen|_applet3AID|  |
   ```

   Case 1: If the SELECT APPLET returns SW1 SW2 == 90 00, then continue with Step 3.

   Case 2: If the SELECT APPLET does not return SW1 SW2 == 90 00, then print
   "The applet cannot be selected. VERIFY PIN (PIN verification required, PIN has already been verified, incorrect PIN code specified) cannot be tested."
   End current test.

3. (Pre) Select the object whose object ID is _object31ID:

Issue the following SELECT OBJECT APDU:

```
00 A4 02 00|02|_object31ID|  |
```

Case 1: If the SELECT OBJECT returns SW1 SW2 == 90 00, then continue with Step 4.

Case 2: If the SELECT OBJECT does not return SW1 SW2 == 90 00, then print
"The object cannot be selected. VERIFY PIN (PIN verification required, PIN has already been verified, incorrect PIN code specified) cannot be tested."
End current test.

4. (Pre) Issue the following VERIFY PIN APDU:

```
00 20 00 00|08|_goodPIN|  |
```

Case 1: If the VERIFY PIN returns SW1 SW2 == 90 00, then continue with Step 5.

Case 2: If the VERIFY PIN does not return SW1 SW2 == 90 00, then print
"The preliminary VERIFY PIN was unsuccessful. VERIFY PIN (PIN verification required, PIN has already been verified, incorrect PIN code specified) cannot be tested."
End current test.

5. Issue the following VERIFY PIN APDU:

```
00 20 00 00|08|_badPIN|  |
```

Verification Goal:
To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 90 00 (successful execution).

2. The services of the currently selected object can be performed.

Perform this verification by attempting to read a value in the currently selected object.

Verification and Reporting Scenario:

1. **Case 1**: If the VERIFY PIN APDU returns response code == 90 00, then:

   Issue the following READ BUFFER APDU:

   ```
   80 52 00 02|02|02 04|  |
   ```

   Case 1.1: If the READ BUFFER APDU returns response code == 90 00 or 61 04, then print
   "VERIFY PIN tested using valid parameters (PIN verification required, PIN has already been verified, incorrect PIN code specified) has been verified because a subsequent READ BUFFER was allowed to read the currently selected object.
   Status: Test 6.6 Passed."

Case 1.2: If the READ BUFFER APDU does not return response code == 90 00 or 61 04, then print

"VERIFY PIN tested using valid parameters (PIN verification required, PIN has already been verified, incorrect PIN code specified) has not been verified because a subsequent READ BUFFER was not allowed to read the currently selected object.
Status: Test 6.6 Failed."

Case 2: If the VERIFY PIN APDU does not return response code == 90 00, then print

"VERIFY PIN tested using valid parameters (PIN verification required, PIN has already been verified, incorrect PIN code specified) has not been verified because the command did not return the correct response code.
Status: Test 6.6 Failed."

Test for Assertion 6.7

The APDU is tested using valid parameters (PIN verification not required, no PIN code specified).

Instantiation Scenario

1. (Pre) Construct the starting state for VERIFY PIN.

2. (Pre) Select the applet whose AID is _applet1AID:

   Issue the following SELECT APPLET APDU:

   ```
   00 A4 04 00|_applet1AIDLen|_applet1AID|  |
   ```

   Case 1: If the SELECT APPLET returns SW1 SW2 == 90 00, then continue with Step 3.

   Case 2: If the SELECT APPLET does not return SW1 SW2 == 90 00, then print
   "The applet cannot be selected. VERIFY PIN (PIN verification not required, no PIN code specified) cannot be tested."
   End current test.

3. (Pre) Select the object whose object ID is _object11ID:

   Issue the following SELECT OBJECT APDU:

   ```
   00 A4 02 00|02|_object11ID|  |
   ```

   Case 1: If the SELECT OBJECT returns SW1 SW2 == 90 00, then continue with Step 4.

   Case 2: If the SELECT OBJECT does not return SW1 SW2 == 90 00, then print
   "The object cannot be selected. VERIFY PIN (PIN verification not required, no PIN code specified) cannot be tested."
   End current test.

4. Issue the following VERIFY PIN APDU:

   ```
   00 20 00 00|00|  |  |
   ```

Verification Goal:
To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 6A 88 (No PIN code defined).

2. The services of the currently selected object can be performed.

Perform this verification by attempting to read a value in the currently selected object.

Verification and Reporting Scenario:

1. **Case 1**: If the VERIFY PIN APDU returns response code == 6A 88, then:

   Issue the following READ BUFFER APDU:

   ```
   80 52 00 02|02|02 04| |
   ```

   Case 1.1: If the READ BUFFER APDU returns response code == 90 00 or 61 04, then print
   "VERIFY PIN tested using valid parameters (PIN verification not required, no PIN code specified) has been verified because a subsequent READ BUFFER was allowed to read the currently selected object.
   Status: Test 6.7 Passed."

   Case 1.2: If the READ BUFFER APDU does not return response code == 90 00 or 61 04, then print
   "VERIFY PIN tested using valid parameters (PIN verification not required, no PIN code specified) has not been verified because a subsequent READ BUFFER was not allowed to read the currently selected object.
   Status: Test 6.7 Failed."

   Case 2: If the VERIFY PIN APDU does not return response code == 6A 88, then print
   "VERIFY PIN tested using valid parameters (PIN verification not required, no PIN code specified) has not been verified because the command did not return the correct response code.
   Status: Test 6.7 Failed."

Test for Assertion 6.8

The APDU is tested using valid parameters (PIN verification not required, PIN code specified).

Instantiation Scenario

1. (Pre) Construct the starting state for VERIFY PIN.

2. (Pre) Select the applet whose AID is _applet1AID:

   Issue the following SELECT APPLET APDU:

   ```
   00 A4 04 00|_applet1AIDLen|_applet1AID| |
   ```

   Case 1: If the SELECT APPLET returns SW1 SW2 == 90 00, then continue with Step 3.

Case 2: If the SELECT APPLET does not return SW1 SW2 == 90 00, then print
"The applet cannot be selected. VERIFY PIN (PIN verification not required, PIN code specified) cannot be tested."
End current test.

3. (Pre) Select the object whose object ID is _object11ID:

   Issue the following SELECT OBJECT APDU:
   00 A4 02 00|02|_object11ID| |

   Case 1: If the SELECT OBJECT returns SW1 SW2 == 90 00, then continue with Step 4.

   Case 2: If the SELECT OBJECT does not return SW1 SW2 == 90 00, then print
   "The object cannot be selected. VERIFY PIN (PIN verification not required, PIN code specified) cannot be tested."
   End current test.

4. Issue the following VERIFY PIN APDU:
   00 20 00 00|08|_goodPIN| |

Verification Goal:
To verify the expected results:
1. The APDU returns
   - SW1 SW2 == 6A 88 (No PIN code defined).

2. The services of the currently selected object can be performed.

Perform this verification by attempting to read a value in the currently selected object.

Verification and Reporting Scenario:
1. **Case 1**: If the VERIFY PIN APDU returns response code == 6A 88, then:

   Issue the following READ BUFFER APDU:
   80 52 00 02|02|02 04| |

   Case 1.1: If the READ BUFFER APDU returns response code == 90 00 or 61 04, then print
   "VERIFY PIN tested using valid parameters (PIN verification not required, PIN code specified) has been verified because a subsequent READ BUFFER was allowed to read the currently selected object.
   Status: Test 6.8 Passed."

   Case 1.2: If the READ BUFFER APDU does not return response code == 90 00 or 61 04, then print
   "VERIFY PIN tested using valid parameters (PIN verification not required, PIN code specified) has not been verified because a subsequent READ BUFFER was not allowed to read the currently selected object.

Status: Test 6.8 Failed."

Case 2: If the VERIFY PIN APDU does not return response code == 6A 88, then print
"VERIFY PIN tested using valid parameters (PIN verification not required, PIN code specified) has not been verified because the command did not return the correct response code.
Status: Test 6.8 Failed."

Test for Assertion 6.9

The APDU is tested using a blocked PIN (correct PIN code specified).

Instantiation Scenario

1. (Pre) Construct the starting state for VERIFY PIN.

2. (Pre) Select the applet whose AID is _applet3AID:

   Issue the following SELECT APPLET APDU:

   ```
   00 A4 04 00|_applet3AIDLen|_applet3AID| |
   ```

   Case 1: If the SELECT APPLET returns SW1 SW2 == 90 00, then continue with Step 3.

   Case 2: If the SELECT APPLET does not return SW1 SW2 == 90 00, then print
   "The applet cannot be selected. VERIFY PIN (PIN verification required, PIN not yet verified, correct PIN code specified) cannot be tested."
   End current test.

3. (Pre) Select the object whose object ID is _object31ID:

   Issue the following SELECT OBJECT APDU:

   ```
   00 A4 02 00|02|_object31ID| |
   ```

   Case 1: If the SELECT OBJECT returns SW1 SW2 == 90 00, then continue with Step 4.

   Case 2: If the SELECT OBJECT does not return SW1 SW2 == 90 00, then print
   "The object cannot be selected. VERIFY PIN (PIN verification required, PIN not yet verified, correct PIN code specified) cannot be tested."
   End current test.

4. (Pre) Issue the following VERIFY PIN APDU:

   ```
   00 20 00 00|08|_badPIN| |
   ```

   Case 1: If the VERIFY PIN returns SW1 SW2 == 63 C2, then continue with Step 5.

   Case 2: If the VERIFY PIN does not return SW1 SW2 == 63 C2, then print

"The first preliminary VERIFY PIN was unsuccessful. VERIFY PIN using a blocked PIN cannot be tested."

5. (Pre) Issue the following VERIFY PIN APDU:

```
00 20 00 00|08|_badPIN| |
```

Case 1: If the VERIFY PIN returns SW1 SW2 == 63 C1, then continue with Step 6.

Case 2: If the VERIFY PIN does not return SW1 SW2 == 63 C1, then print
"The second preliminary VERIFY PIN was unsuccessful. VERIFY PIN using a blocked PIN cannot be tested."

6. (Pre) Issue the following VERIFY PIN APDU:

```
00 20 00 00|08|_badPIN| |
```

Case 1: If the VERIFY PIN returns SW1 SW2 == 63 C0, then continue with Step 7.

Case 2: If the VERIFY PIN does not return SW1 SW2 == 63 C0, then print
"The preliminary VERIFY PIN was unsuccessful. VERIFY PIN using a blocked PIN cannot be tested."

7. Issue the following VERIFY PIN APDU:

```
00 20 00 00|08|_goodPIN| |
```

Verification Goal:
To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 69 83 (PIN code blocked).

2. The services of the currently selected object cannot be performed.

Perform this verification by attempting to read a value in the currently selected object.

Verification and Reporting Scenario:

1. **Case 1**: If the VERIFY PIN APDU returns response code == 69 83, then:

   Issue the following READ BUFFER APDU:

   ```
   80 52 00 02|02|02 04| |
   ```

   Case 1.1: If the READ BUFFER APDU returns response code == 69 82, then print
   "VERIFY PIN tested using a blocked PIN has been verified because a subsequent READ BUFFER was not allowed to read the currently selected object.
   Status: Test 6.9 Passed."

   Case 1.2: If the READ BUFFER APDU does not return response code == 69 82, then print

"VERIFY PIN tested using a blocked PIN has not been verified because a subsequent READ BUFFER returned an incorrect return code.
Status: Test 6.9 Passed."

Case 2: If the VERIFY PIN APDU does not return response code == 69 83, then print
"VERIFY PIN tested using a blocked PIN has not been verified because the command did not return the correct response code.
Status: Test 6.9 Failed."

Test for Assertion 6.10

The APDU is tested using an invalid L_c.

This test is not part of the current version of the GSC-IS Version 2.1 Card Edge Interface Test Suite.

7. READ BUFFER

Starting State for Each Test:

1. A card that claims to implement the GSC-IS, Version 2.1, is in a reader.

2. The card contains an applet whose AID is _applet1AID. This AID has length _applet1AIDLen.

3. The applet whose AID is _applet1AID contains an object with object ID _object11ID. This object
   - contains _tag1 in byte 02, 04 in byte 03, _tag2 in byte 04, and 08 in byte 05 of its T-buffer
   - contains _existingValue1 in bytes 02 through 05, and _existingValue2 in bytes 06-0D of its V-buffer
   - has access control rule ACR_ALWAYS for all services.

4. The card contains a second applet whose AID is _applet2AID. This AID has length _applet2AIDLen.

5. The applet whose AID is _applet2AID contains an object with object ID _object21ID. This object
   - contains _tag8 in byte 02, and 04 in byte 03, of its T-buffer
   - contains _existingValue8 in bytes 02 through 05 of its V-buffer
   - has access control rule ACR_XAUTH_AND_PIN for all services.

Test for Assertion 7.1

The APDU is tested using valid parameters (T-buffer).

Instantiation Scenario

1. (Pre) Construct the starting state for READ BUFFER.

2. (Pre) Select the applet whose AID is _applet1AID:

   Issue the following SELECT APPLET APDU:

   00 A4 04 00│_applet1AIDLen│_applet1AID│ │

   Case 1: If the SELECT APPLET returns SW1 SW2 == 90 00, then continue with Step 3.

   Case 2: If the SELECT APPLET does not return SW1 SW2 == 90 00, then print
   "The applet cannot be selected. READ BUFFER cannot be tested."
   End current test.

3. (Pre) Select the object whose object ID is _object11ID:

   Issue the following SELECT OBJECT APDU:

   00 A4 02 00│02│_object11ID│ │

Case 1: If the SELECT OBJECT returns SW1 SW2 == 90 00, then continue with Step 4.

Case 2: If the SELECT OBJECT does not return SW1 SW2 == 90 00, then print
"The object cannot be selected. READ BUFFER cannot be tested."
End current test.

4. Issue the following READ BUFFER APDU:

   80 52 00 02│02│01 04│ │

Verification Goal:
To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 90 00 (successful execution)
   - the response data field == _tag1 04 _tag2 08.

   or
   - SW1 SW2 == 61 04 (normal processing, data still available)
   - the available data == _tag1 04 _tag2 08.

Verification and Reporting Scenario:

1. **Case 1**: If the READ BUFFER APDU returns response code == 90 00, then verify that the response data field contains the correct T-buffer entries:

   Case 1.1: If
   - the response data field == _tag1 04 _tag2 08

   then print
   "READ BUFFER tested using valid parameters (T-buffer) has been verified.
   Status: Test 7.1 Passed."

   Case 1.2: If it is not true that
   - the response data field == _tag1 04 _tag2 08

   then print
   "READ BUFFER tested using valid parameters (T-buffer) has not been verified because the APDU did not return the correct T-buffer entries.
   Status: Test 7.1 Failed."

   Case 2: If the READ BUFFER APDU returns response code == 61 04, then issue the following GET RESPONSE:

   00 C0 00 00│ │ │04│

   Case 2.1: If the GET RESPONSE returns response code 90 00, then verify that the response data field contains the correct T-buffer entries:

   Case 2.1.1: If
   - the response data field == _tag1 04 _tag2 08

   then print
   "READ BUFFER tested using valid parameters (T-buffer) has been verified.
   Status: Test 7.1 Passed."

   Case 2.1.2: If it is not true that

- the response data field == _tag1 04 _tag2 08

then print

"READ BUFFER tested using valid parameters (T-buffer) has not been verified because a subsequent GET RESPONSE did not return the correct T-buffer entries.
Status: Test 7.1 Failed."

Case 2.2: If the GET RESPONSE does not return response code 90 00, then print

"READ BUFFER tested using valid parameters (T-buffer) has not been verified because a subsequent GET RESPONSE did not return the correct response code.
Status: Test 7.1 Undetermined."

Case 3: If the READ BUFFER APDU does not return response code == 90 00 or 61 04, then print

"READ BUFFER tested using valid parameters (T-buffer) has not been verified because the APDU did not return the correct response code.
Status: Test 7.1 Failed."

Test for Assertion 7.2

The APDU is tested using valid parameters (V-buffer).

Instantiatio Scenario

1. (Pre) Construct the starting state for READ BUFFER.

2. (Pre) Select the applet whose AID is _applet1AID:

   Issue the following SELECT APPLET APDU:

   00 A4 04 00|_applet1AIDLen|_applet1AID| |

   Case 1: If the SELECT APPLET returns SW1 SW2 == 90 00, then continue with Step 3.

   Case 2: If the SELECT APPLET does not return SW1 SW2 == 90 00, then print
   "The applet cannot be selected. READ BUFFER cannot be tested."
   End current test.

3. (Pre) Select the object whose object ID is _object11ID:

   Issue the following SELECT OBJECT APDU:

   00 A4 02 00|02|_object11ID| |

   Case 1: If the SELECT OBJECT returns SW1 SW2 == 90 00, then continue with Step 4.

   Case 2: If the SELECT OBJECT does not return SW1 SW2 == 90 00, then print
   "The object cannot be selected. READ BUFFER cannot be tested."
   End current test.

4. Issue the following READ BUFFER APDU:

```
80 52 00 02|02|02 08| |
```

Verification Goal:

To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 90 00 (successful execution)
   - the response data field == _existingValue1 _existingValue2.

   or
   - SW1 SW2 == 61 08 (normal processing, data still available)
   - the available data == _existingValue1 _existingValue2.

Verification and Reporting Scenario:

1. **Case 1**: If the READ BUFFER APDU returns response code == 90 00, then verify that the response data field contains the correct V-buffer entry:

   Case 1.1: If
   - the response data field == _existingValue1 _existingValue2

   then print

   "READ BUFFER tested using valid parameters (V-buffer) has been verified.
   Status: Test 7.2 Passed."

   Case 1.2: If it is not true that
   - the response data field == _existingValue1 _existingValue2

   then print

   "READ BUFFER tested using valid parameters (V-buffer) has not been verified because the APDU did not return the correct V-buffer entry.
   Status: Test 7.2 Failed."

   Case 2: If the READ BUFFER APDU returns response code == 61 08, then issue the following GET RESPONSE:

   ```
   00 C0 00 00| | |08|
   ```

   Case 2.1: If the GET RESPONSE returns response code 90 00, then verify that the response data field contains the correct V-buffer entry:

   Case 2.1.1: If
   - the response data field == _existingValue1 _existingValue2

   then print

   "READ BUFFER tested using valid parameters (V-buffer) has been verified.
   Status: Test 7.2 Passed."

   Case 2.1.2: If it is not true that
   - the response data field == _existingValue1 _existingValue2

   then print

   "READ BUFFER tested using valid parameters (V-buffer) has not been verified because a subsequent GET RESPONSE did not return the correct V-buffer entry.

Status: Test 7.2 Failed."

Case 2.2: If the GET RESPONSE does not return response code 90 00, then print
"READ BUFFER tested using valid parameters (V-buffer) has not been verified because a subsequent GET RESPONSE did not return the correct response code.
Status: Test 7.2 Undetermined."

Case 3: If the READ BUFFER APDU does not return response code == 90 00 or 61 0C, then print
"READ BUFFER tested using valid parameters (V-buffer) has not been verified because the APDU did not return the correct response code.
Status: Test 7.2 Failed."

Test for Assertion 7.3

The APDU is tested using an incorrect parameter L_c (V-buffer).

This test is not part of the current version of the GSC-IS Version 2.1 Card Edge Interface Test Suite.

Test for Assertion 7.4

The APDU is tested using an invalid parameter in the Data Field.

Instantiation Scenario

1. (Pre) Construct the starting state for READ BUFFER.

2. (Pre) Select the applet whose AID is _applet1AID:

   Issue the following SELECT APPLET APDU:
   00 A4 04 00│_applet1AIDLen│_applet1AID│ │

   Case 1: If the SELECT APPLET returns SW1 SW2 == 90 00, then continue with Step 3.

   Case 2: If the SELECT APPLET does not return SW1 SW2 == 90 00, then print
   "The applet cannot be selected. READ BUFFER cannot be tested."
   End current test.

3. (Pre) Select the object whose object ID is _object11ID:

   Issue the following SELECT OBJECT APDU:
   00 A4 02 00│02│_object11ID│ │

   Case 1: If the SELECT OBJECT returns SW1 SW2 == 90 00, then continue with Step 4.

   Case 2: If the SELECT OBJECT does not return SW1 SW2 == 90 00, then print
   "The object cannot be selected. READ BUFFER cannot be tested."

End current test.

4. Issue the following READ BUFFER APDU:

   ```
   80 52 00 02|02|FE 02| |
   ```

Verification Goal:
To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 6A 88 (invalid parameter in the Data Field).

Verification and Reporting Scenario:

1. **Case 1**: If the READ BUFFER APDU returns response code == 6A 88, then print
   "READ BUFFER tested using an invalid parameter in the Data Field has been verified.
   Status: Test 7.4 Passed."

   Case 2: If the READ BUFFER APDU does not return response code == 6A 88, then print
   "READ BUFFER tested using an invalid parameter in the Data Field has not been verified because the APDU did not return the correct response code.
   Status: Test 7.4 Failed."

Test for Assertion 7.5

The APDU is tested using an invalid P1 or P2 parameter (V-buffer).

Instantiatio Scenario

1. (Pre) Construct the starting state for READ BUFFER.

2. (Pre) Select the applet whose AID is _applet1AID:

   Issue the following SELECT APPLET APDU:

   ```
   00 A4 04 00|_applet1AIDLen|_applet1AID| |
   ```

   Case 1: If the SELECT APPLET returns SW1 SW2 == 90 00, then continue with Step 3.

   Case 2: If the SELECT APPLET does not return SW1 SW2 == 90 00, then print
   "The applet cannot be selected. READ BUFFER cannot be tested."
   End current test.

3. (Pre) Select the object whose object ID is _object11ID:

   Issue the following SELECT OBJECT APDU:

   ```
   00 A4 02 00|02|_object11ID| |
   ```

   Case 1: If the SELECT OBJECT returns SW1 SW2 == 90 00, then continue with Step 4.

   Case 2: If the SELECT OBJECT does not return SW1 SW2 == 90 00, then print

"The object cannot be selected. READ BUFFER cannot be tested."
End current test.

4. Issue the following READ BUFFER APDU:

```
80 52 00 FE|02|02 06|  |
```

Verification Goal:

To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 6A 80 (invalid P1 or P2 parameter).

Verification and Reporting Scenario:

1. **Case 1**: If the READ BUFFER APDU returns response code == 6A 80, then print
   "READ BUFFER tested using an invalid P1 or P2 parameter (V-buffer) has been verified.
   Status: Test 7.5 Passed."

   Case 2: If the READ BUFFER APDU does not return response code == 6A 80, then print
   "READ BUFFER tested using an invalid P1 or P2 parameter (V-buffer) has not been verified because the APDU did not return the correct response code.
   Status: Test 7.5 Failed."

Test for Assertion 7.6

The APDU is tested where the security status of the currently selected object is not satisfied (V-buffer).

Instantiation Scenario

1. (Pre) Construct the starting state for READ BUFFER.

2. (Pre) Select the applet whose AID is _applet2AID:

   Issue the following SELECT APPLET APDU:

   ```
   00 A4 04 00|_applet2AIDLen|_applet2AID|  |
   ```

   Case 1: If the SELECT APPLET returns SW1 SW2 == 90 00, then continue with Step 3.

   Case 2: If the SELECT APPLET does not return SW1 SW2 == 90 00, then print
   "The applet cannot be selected. READ BUFFER cannot be tested."
   End current test.

3. (Pre) Select the object whose object ID is _object21ID:

   Issue the following SELECT OBJECT APDU:

   ```
   00 A4 02 00|02|_object21ID|  |
   ```

   Case 1: If the SELECT OBJECT returns SW1 SW2 == 90 00, then continue with Step 4.

Case 2: If the SELECT OBJECT does not return SW1 SW2 == 90 00, then print
"The object cannot be selected. READ BUFFER cannot be tested."
End current test.

4. Issue the following READ BUFFER APDU:

   80 52 00 02|02|02 04| |.

Note: The security status of the selected object is deliberately not satisfied.

<u>Verification Goal</u>:
To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 69 82 (security status not satisfied).

<u>Verification and Reporting Scenario</u>:

2. **Case 1**: If the READ BUFFER APDU returns response code == 69 82, then print
   "READ BUFFER tested using an invalid P1 or P2 parameter (V-buffer) has been verified.
   Status: Test 7.6 Passed."

   Case 2: If the READ BUFFER APDU does not return response code == 69 82, then print
   "READ BUFFER tested using an invalid P1 or P2 parameter (V-buffer) has not been verified because the APDU did not return the correct response code.
   Status: Test 7.6 Failed."

Test for Assertion 7.7

The APDU is tested when no container is currently selected.

<u>Instantiation Scenario</u>

1. (Pre) Construct the starting state for READ BUFFER.

2. (Pre) Select the applet whose AID is _applet1AID:

   Issue the following SELECT APPLET APDU:

   00 A4 04 00|_applet1AIDLen|_applet1AID| |

   Case 1: If the SELECT APPLET returns SW1 SW2 == 90 00, then continue with Step 3.

   Case 2: If the SELECT APPLET does not return SW1 SW2 == 90 00, then print
   "The applet cannot be selected. READ BUFFER cannot be tested."
   End current test.

3. Issue the following READ BUFFER APDU:

   80 52 00 02|02|02 04| |.

<u>Verification Goal</u>:

To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 6A 82 (applet or object not found).

<u>Verification and Reporting Scenario</u>:

1. **Case 1**: If the READ BUFFER APDU returns response code == 6A 82, then print
   "READ BUFFER tested when no container is currently selected has been verified.
   Status: Test 7.7 Passed."

   Case 2: If the READ BUFFER APDU does not return response code == 6A 82, then print
   "READ BUFFER tested when no container is currently selected has not been verified because the APDU did not return the correct response code.
   Status: Test 7.7 Failed."

8. UPDATE BUFFER

Starting State for Each Test:

1. A card that claims to implement the GSC-IS, Version 2.1, is in a reader.

2. The card contains an applet whose AID is _applet5AID. This AID has length _applet5AIDLen.

3. The applet whose AID is _applet5AID contains an object with object ID _object51ID. This object
   - has T-buffer: 000C _tag3 04 _tag4 04 _tag5 04 _tag6 04 _tag7 04
   - has V-buffer: 0016 _existingValue3 _existingValue4 _existingValue5 _existingValue6 _existingValue7
   - has access control rule ACR_ALWAYS for all services.

4. The card contains a second applet whose AID is _applet4AID. This AID has length _applet4AIDLen.

5. The applet whose AID is _applet4AID contains an object with object ID _object41ID. This object
   - has T-buffer: 0006 _tag10 04 _tag11 04
   - has V-buffer: 0006 _existingValue10 _existingValue11
   - has access control rule ACR_ALWAYS for the read service, and ACR_XAUTH for the update service.

Test for Assertion 8.1

The APDU is tested using valid parameters (T-buffer).

Instantiation Scenario

1. (Pre) Construct the starting state for UPDATE BUFFER.

2. (Pre) Select the applet whose AID is _applet5AID:

   Issue the following SELECT APPLET APDU:

   ```
   00 A4 04 00|_applet5AIDLen|_applet5AID| |
   ```

   Case 1: If the SELECT APPLET returns SW1 SW2 == 90 00, then continue with Step 3.

   Case 2: If the SELECT APPLET does not return SW1 SW2 == 90 00, then print
   "The applet cannot be selected. UPDATE BUFFER cannot be tested."
   End current test.

3. (Pre) Select the object whose object ID is _object51ID:

   Issue the following SELECT OBJECT APDU:

   ```
   00 A4 02 00|02|_object51ID| |
   ```

Case 1: If the SELECT OBJECT returns SW1 SW2 == 90 00, then continue with Step 4.

Case 2: If the SELECT OBJECT does not return SW1 SW2 == 90 00, then print
"The object cannot be selected. UPDATE BUFFER cannot be tested."
End current test.

4. Issue the following UPDATE BUFFER APDU:

```
80 58 00 02|02|01 _newTag|  |
```

Verification Goal:
To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 90 00 (successful execution).

2. The block of data in the T-buffer specified to be updated now == _newTag.

Perform this verification by issuing a READ BUFFER.

Verification and Reporting Scenario:

1. **Case 1**: If the UPDATE BUFFER APDU returns response code == 90 00, then verify that the block of data in the T-buffer specified to be updated now == _newTag:

   Issue the following READ BUFFER APDU:

   ```
   80 52 00 02|02|01 01|  |
   ```

   Case 1.1: If the READ BUFFER returns response code 90 00 and the response data field == _newTag, then print
   "UPDATE BUFFER using valid parameters (T-buffer) has been verified.
   Status: Test 8.1 Passed."

   Case 1.2: If the READ BUFFER returns response code 90 00 and the response data field /= _newTag, then print
   "UPDATE BUFFER using valid parameters (T-buffer) has not been verified because the data was not updated correctly.
   Status: Test 8.1 Failed."

   Case 1.3: If the READ BUFFER returns response code 61 01, then issue the following GET RESPONSE:

   ```
   00 C0 00 00|  |  |01|
   ```

   Case 1.3.1: If the GET RESPONSE returns response code 90 00, then verify that the response data field contains the correct T-buffer entry:

   Case 1.3.1.1: If the response data field == _newTag, then print
   "UPDATE BUFFER tested using valid parameters (T-buffer) has been verified.
   Status: Test 8.1 Passed."

Case 1.3.1.2: If the response data field /= _newTag, then print
"UPDATE BUFFER tested using valid parameters (T-buffer) has not been verified because the data was not updated correctly.
Status: Test 8.1 Failed."

Case 1.3.2: If the GET RESPONSE does not return response code 90 00, then print:
"UPDATE BUFFER tested using valid parameters (T-buffer) has not been verified because a subsequent GET RESPONSE did not return the correct response code.
Status: Test 8.1 Undetermined.

Case 1.4: If the READ BUFFER does not return response code 90 00 or 61 01, then print
"UPDATE BUFFER using valid parameters (T-buffer) has not been verified because a subsequent READ BUFFER did not return the correct response code.
Status: Test 8.1 Undetermined."

Case 2: If the UPDATE BUFFER APDU does not return response code == 90 00, then print
"UPDATE BUFFER tested using valid parameters (T-buffer) has not been verified because the APDU did not return the correct response code.
Status: Test 8.1 Failed."

Test for Assertion 8.2

The APDU is tested using valid parameters (V-buffer).

Instantiation Scenario

1. (Pre) Construct the starting state for UPDATE BUFFER.

2. (Pre) Select the applet whose AID is _applet5AID:

   Issue the following SELECT APPLET APDU:

   ```
   00 A4 04 00|_applet5AIDLen|_applet5AID|  |
   ```

   Case 1: If the SELECT APPLET returns SW1 SW2 == 90 00, then continue with Step 3.

   Case 2: If the SELECT APPLET does not return SW1 SW2 == 90 00, then print
   "The applet cannot be selected. UPDATE BUFFER cannot be tested."
   End current test.

3. (Pre) Select the object whose object ID is _object51ID:

   Issue the following SELECT OBJECT APDU:

   ```
   00 A4 02 00|02|_object51ID|  |
   ```

Case 1: If the SELECT OBJECT returns SW1 SW2 == 90 00, then continue with Step 4.

Case 2: If the SELECT OBJECT does not return SW1 SW2 == 90 00, then print
"The object cannot be selected. UPDATE BUFFER cannot be tested."
End current test.

4. Issue the following UPDATE BUFFER APDU:
   80 58 00 06|05|02 _newValue1| |

Verification Goal:
To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 90 00 (successful execution).
2. The block of data in the V-buffer specified to be updated now == _newValue1.

Perform this verification by issuing a READ BUFFER.

Verification and Reporting Scenario:

1. **Case 1**: If the UPDATE BUFFER APDU returns response code == 90 00, then verify that the block of data in the V-buffer specified to be updated now == _newTag:

Issue the following READ BUFFER APDU:
80 52 00 06|02|02 04| |

Case 1.1: If the READ BUFFER returns response code 90 00 and the response data field == _newValue1, then print
"UPDATE BUFFER using valid parameters (V-buffer) has been verified.
Status: Test 8.2 Passed."

Case 1.2: If the READ BUFFER returns response code 90 00 and the response data field /= _newValue1, then print
"UPDATE BUFFER using valid parameters (V-buffer) has not been verified because the data was not updated correctly.
Status: Test 8.2 Failed."

Case 1.3: If the READ BUFFER returns response code 61 04, then issue the following GET RESPONSE:
00 C0 00 00| | |04|

Case 1.3.1: If the GET RESPONSE returns response code 90 00, then verify that the response data field contains the correct V-buffer entry:

Case 1.3.1.1: If the response data field == _newValue1, then print
"UPDATE BUFFER tested using valid parameters (V-buffer) has been verified.
Status: Test 8.2 Passed."

Case 1.3.1.2: If the response data field /= _newValue1, then print

"UPDATE BUFFER tested using valid parameters (V-buffer) has not been verified because the data was not updated correctly.
Status: Test 8.2 Failed."

Case 1.3.2: If the GET RESPONSE does not return response code 90 00, then print:

"UPDATE BUFFER tested using valid parameters (V-buffer) has not been verified because a subsequent GET RESPONSE did not return the correct response code.
Status: Test 8.2 Undetermined.

Case 1.4: If the READ BUFFER does not return response code 90 00 or 61 04, then print

"UPDATE BUFFER using valid parameters (V-buffer) has not been verified because a subsequent READ BUFFER did not return the correct response code.
Status: Test 8.2 Undetermined."

Case 2: If the UPDATE BUFFER APDU does not return response code == 90 00, then print

"UPDATE BUFFER tested using valid parameters (V-buffer) has not been verified because the APDU did not return the correct response code.
Status: Test 8.2 Failed."

Test for Assertion 8.3

The APDU is tested using an invalid parameter L_c (V-buffer).

This test is not part of the current version of the GSC-IS Version 2.1 Card Edge Interface Test Suite.

Test for Assertion 8.4

The APDU is tested using an invalid parameter in the data field.

Instantiation Scenario

1. (Pre) Construct the starting state for UPDATE BUFFER.

2. (Pre) Select the applet whose AID is _applet5AID:

   Issue the following SELECT APPLET APDU:

   00 A4 04 00|_applet5AIDLen|_applet5AID| |

   Case 1: If the SELECT APPLET returns SW1 SW2 == 90 00, then continue with Step 3.

   Case 2: If the SELECT APPLET does not return SW1 SW2 == 90 00, then print

"The applet cannot be selected. UPDATE BUFFER cannot be tested."
End current test.

3. (Pre) Select the object whose object ID is _object51ID:

   Issue the following SELECT OBJECT APDU:

   ```
   00 A4 02 00|02|_object51ID|  |
   ```

   Case 1: If the SELECT OBJECT returns SW1 SW2 == 90 00, then continue with Step 4.

   Case 2: If the SELECT OBJECT does not return SW1 SW2 == 90 00, then print
   "The object cannot be selected. UPDATE BUFFER cannot be tested."
   End current test.

4. Issue the following UPDATE BUFFER APDU:

   ```
   80 58 00 0E|05|FE _newValue2|  |
   ```

Verification Goal:
To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 6A 80 or 6A 88 (invalid parameters in command field or no corresponding buffer).

Verification and Reporting Scenario:

1. **Case 1**: If the UPDATE BUFFER APDU returns response code == 6A 80 or 6A 88, then print
   "UPDATE BUFFER using an invalid parameter in the data field has been verified.
   Status: Test 8.4 Passed.

   Case 2: If the UPDATE BUFFER APDU does not return response code == 6A 80 or 6A 88, then print
   "UPDATE BUFFER tested using an invalid parameter in the data field has not been verified because the APDU did not return the correct response code.
   Status: Test 8.4 Failed."

Test for Assertion 8.5

The APDU is tested using an invalid P1 or P2 parameter (V-buffer).

Instantiation Scenario

1. (Pre) Construct the starting state for UPDATE BUFFER.

2. (Pre) Select the applet whose AID is _applet5AID:

   Issue the following SELECT APPLET APDU:

   ```
   00 A4 04 00|_applet5AIDLen|_applet5AID|  |
   ```

Case 1: If the SELECT APPLET returns SW1 SW2 == 90 00, then continue with Step 3.

Case 2: If the SELECT APPLET does not return SW1 SW2 == 90 00, then print
"The applet cannot be selected. UPDATE BUFFER cannot be tested."
End current test.

3. (Pre) Select the object whose object ID is _object51ID:

   Issue the following SELECT OBJECT APDU:

   ```
   00 A4 02 00|02|_object51ID| |
   ```

   Case 1: If the SELECT OBJECT returns SW1 SW2 == 90 00, then continue with Step 4.

   Case 2: If the SELECT OBJECT does not return SW1 SW2 == 90 00, then print
   "The object cannot be selected. UPDATE BUFFER cannot be tested."
   End current test.

4. Issue the following UPDATE BUFFER APDU:

   ```
   80 58 00 FE|05|01 _newValue3| |
   ```

Verification Goal:
To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 6A 86 (invalid P1-P2).

Verification and Reporting Scenario:

1. **Case 1**: If the UPDATE BUFFER APDU returns response code == 6A 86, then print
   "UPDATE BUFFER using an invalid P1 or P2 parameter has been verified.
   Status: Test 8.5 Passed.

   Case 2: If the UPDATE BUFFER APDU does not return response code == 6A 86, then print
   "UPDATE BUFFER tested using an invalid P1 or P2 parameter has not been verified because the APDU did not return the correct response code.
   Status: Test 8.5 Failed."

Test for Assertion 8.6

The APDU is tested where the security status of the currently selected object is not satisfied (V-buffer).

Instantiation Scenario

1. (Pre) Construct the starting state for UPDATE BUFFER.

2. (Pre) Select the applet whose AID is _applet4AID:

Issue the following SELECT APPLET APDU:

```
00 A4 04 00|_applet4AIDLen|_applet4AID|  |
```

Case 1: If the SELECT APPLET returns SW1 SW2 == 90 00, then continue with Step 3.

Case 2: If the SELECT APPLET does not return SW1 SW2 == 90 00, then print
"The applet cannot be selected. UPDATE BUFFER cannot be tested."
End current test.

3. (Pre) Select the object whose object ID is _object41ID:

Issue the following SELECT OBJECT APDU:

```
00 A4 02 00|02|_object41ID|  |
```

Case 1: If the SELECT OBJECT returns SW1 SW2 == 90 00, then continue with Step 4.

Case 2: If the SELECT OBJECT does not return SW1 SW2 == 90 00, then print
"The object cannot be selected. UPDATE BUFFER cannot be tested."
End current test.

4. Issue the following UPDATE BUFFER APDU:

```
80 58 00 02|05|02 _newValue4|  |
```

Verification Goal:
To verify the expected results:
1. The APDU returns
   - SW1 SW2 == 69 82 (security status not satisfied).

2. The block of data in the V-buffer specified to be updated has not been changed.

Perform this verification by issuing a READ BUFFER.

Verification and Reporting Scenario:
1. **Case 1**: If the UPDATE BUFFER APDU returns response code == 69 82, then verify that the block of data in the V-buffer specified to be updated still has the value _existingValue10:

Issue the following READ BUFFER APDU:

```
80 52 00 02|02|02 04|  |
```

Case 1.1: If the READ BUFFER returns response code 90 00 and the response data field == _existingValue10, then print
"UPDATE BUFFER tested where the security status of the currently selected object is not satisfied has been verified.
Status: Test 8.6 Passed."

Case 1.2: If the READ BUFFER returns response code 90 00 and the response data field /= _existingValue10, then print

"UPDATE BUFFER tested where the security status of the currently selected object is not satisfied has not been verified because the V-buffer entry was changed.
Status: Test 8.6 Failed."

Case 1.3: If the READ BUFFER returns response code 61 04, then issue the following GET RESPONSE:

00 C0 00 00| | |04|

Case 1.3.1: If the GET RESPONSE returns response code 90 00, then verify that the response data field contains the original V-buffer entry:

Case 1.3.1.1: If the response data field == _existingValue10, then print
"UPDATE BUFFER tested where the security status of the currently selected object is not satisfied has been verified.
Status: Test 8.6 Passed."

Case 1.3.1.2: If the response data field /= _existingValue10, then print
"UPDATE BUFFER tested where the security status of the currently selected object is not satisfied has not been verified because the V-buffer entry was changed.
Status: Test 8.6 Failed."

Case 1.3.2: If the GET RESPONSE does not return response code 90 00, then print:
"UPDATE BUFFER tested where the security status of the currently selected object is not satisfied has not been verified because a subsequent GET RESPONSE did not return the correct response code.
Status: Test 8.6 Undetermined.

Case 1.4: If the READ BUFFER does not return response code 90 00 or 61 04, then print
"UPDATE BUFFER tested where the security status of the currently selected object is not satisfied has not been verified because a subsequent READ BUFFER did not return the correct response code.
Status: Test 8.6 Undetermined."

Case 2: If the UPDATE BUFFER APDU does not return response code == 69 82, then print
"UPDATE BUFFER tested where the security status of the currently selected object is not satisfied has not been verified because the APDU did not return the correct response code.
Status: Test 8.6 Failed."

Test for Assertion 8.7

The APDU is tested when no container is currently selected.

Instantiation Scenario

1. (Pre) Construct the starting state for UPDATE BUFFER.

2. (Pre) Select the applet whose AID is _applet5AID:

   Issue the following SELECT APPLET APDU:

   ```
   00 A4 04 00|_applet5AIDLen|_applet5AID| |
   ```

   Case 1: If the SELECT APPLET returns SW1 SW2 == 90 00, then continue with Step 3.

   Case 2: If the SELECT APPLET does not return SW1 SW2 == 90 00, then print
   "The applet cannot be selected. UPDATE BUFFER cannot be tested."
   End current test.

3. Issue the following UPDATE BUFFER APDU:

   ```
   80 58 00 00|05|02 _newValue5| |
   ```

Verification Goal:

To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 6A 82 (applet or object not found).

Verification and Reporting Scenario:

1. **Case 1**: If the UPDATE BUFFER APDU returns response code == 6A 82, then print
   "UPDATE BUFFER tested when no container is currently selected has been verified.
   Status: Test 8.7 Passed."

   Case 2: If the UPDATE BUFFER APDU does not return response code == 6A 82, then print
   "UPDATE BUFFER tested when no container is currently selected has not been verified because the APDU did not return the correct response code.
   Status: Test 8.7 Failed."

9. GET CHALLENGE

Starting State for Each Test:

1. A card that claims to implement the GSC-IS, Version 2.1, is in a reader.

2. The card contains an applet whose AID is _applet1AID. The length of this AID is _applet1AIDLen.

3. Issue the following SELECT APPLET APDU:

   00 A4 04 00|_applet1AIDLen|_applet1AID| |.

Test for Assertion 9.1

The APDU is tested using valid parameters.

Instantiation Scenario

1. (Pre) Construct the starting state for GET CHALLENGE.

2. Issue the following GET CHALLENGE APDU:

   00 84 00 00| | |08|

Verification Goal:
To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 90 00 (successful execution)
   - the 8 byte data field in the response message == a cryptographic challenge.

Perform this verification by inspection.

Verification and Reporting Scenario:

1. **Case 1**: If the GET CHALLENGE APDU returns SW1 SW2 == 90 00, then manually inspect the 8 byte data field in the response message.

   Case 1.1: If the data field contains a valid 8 byte cryptographic challenge, then print
   "GET CHALLENGE tested using valid parameters has been verified.
   Status: Test 9.1 Passed."

   Case 1.2: If the data field does not contain a valid 8 byte cryptographic challenge, then print
   "GET CHALLENGE tested using valid parameters has not been verified by inspection.
   Status: Test 9.1 Failed."

   Case 2: If the GET CHALLENGE APDU does not return SW1 SW2 == 90 00, then print
   "GET CHALLENGE tested using valid parameters did not return the correct status code.
   Status: Test 9.1 Failed."

Test for Assertion 9.2

The APDU is tested where the specified length of the returned challenge is incorrect.

Instantiation Scenario

1. (Pre) Construct the starting state for GET CHALLENGE.

2. Issue the following GET CHALLENGE APDU:

```
00 84 00 00|  |  |01|
```

Verification Goal:

To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 6C 08 (Wrong Length in L_e parameter).

Verification and Reporting Scenario:

1. **Case 1**: If the GET CHALLENGE APDU returns SW1 SW2 == 6C 08, then print

   "GET CHALLENGE tested where the specified length of the returned challenge is incorrect has been verified.
   Status: Test 9.2 Passed."

   Case 2: If the GET CHALLENGE APDU does not return SW1 SW2 == 6C 08, then print

   "GET CHALLENGE tested where the specified length of the returned challenge is incorrect did not return the correct status code.
   Status: Test 9.2 Failed."

Test for Assertion 9.3

The APDU is tested using an invalid P1 or P2 parameter.

Instantiation Scenario

1. (Pre) Construct the starting state for GET CHALLENGE.

2. Issue the following GET CHALLENGE APDU:

```
00 84 00 FE|  |  |08|
```

Verification Goal:

To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 6A 86 (Incorrect P1 or P2 parameter).

Verification and Reporting Scenario:

1. **Case 1**: If the GET CHALLENGE APDU returns SW1 SW2 == 6A 86, then print

   "GET CHALLENGE tested using an invalid P1 or P2 parameter has been verified.
   Status: Test 9.3 Passed."

   Case 2: If the GET CHALLENGE APDU does not return SW1 SW2 == 6A 86, then print

"GET CHALLENGE tested using an invalid P1 or P2 parameter did not return the correct status code.
Status: Test 9.3 Failed."

10. EXTERNAL AUTHENTICATE

Starting State for Each Test:

1. A card that claims to implement the GSC-IS, Version 2.1, is in a reader.

2. The card contains an applet whose AID is _applet4AID. This AID has length _applet4AIDLen.

3. The applet whose AID is _applet4AID contains an object with object ID _object41ID. This object
   - has T-buffer: 0006 _tag10 04 _tag11 04
   - has V-buffer: 000A _existingValue10 _existingValue11
   - has access control rule ACR_ALWAYS for the read service, and ACR_XAUTH for the update service.

4. Select the applet whose AID is _applet4AID:

   Issue the following SELECT APPLET APDU:

   ```
   00 A4 04 00|_applet4AIDLen|_applet4AID|  |
   ```

   Case 1: If the SELECT APPLET returns SW1 SW2 == 90 00, then continue with Step 5.

   Case 2: If the SELECT APPLET does not return SW1 SW2 == 90 00, then print
   "The applet cannot be selected. EXTERNAL AUTHENTICATE cannot be tested."
   End current test.

5. Select the object whose object ID is _object41ID:

   Issue the following SELECT OBJECT APDU:

   ```
   00 A4 02 00|02|_object41ID|  |
   ```

   Case 1: If the SELECT OBJECT returns SW1 SW2 == 90 00, then continue with Step 6.

   Case 2: If the SELECT OBJECT does not return SW1 SW2 == 90 00, then print
   "The object cannot be selected. EXTERNAL AUTHENTICATE cannot be tested."
   End current test.

6. Verify that the object whose objectID is _object41ID does not allow update access:

   Issue the following UPDATE BUFFER APDU:

   ```
   80 58 00 02 05 02 _newValue5
   ```

   Case 1: If the UPDATE BUFFER returns SW1 SW2 == 69 82, then continue with Step 7.

Case 2: If the UPDATE BUFFER does not return SW1 SW2 == 69 82, then print

"It cannot be verified that the currently selected object does not allow update access. EXTERNAL AUTHENTICATE cannot be tested."

End current test.

7. Issue the following GET CHALLENGE APDU:

   00 84 00 00│ │ │08│

   Case 1: If the GET CHALLENGE returns SW1 SW2 == 90 00, then

   Let challenge1000 == the 8 byte data field in the response message.

   Encrypt challenge1000, producing cryptogram1000.

   Continue with Step 2 of the Instantiation Scenario.

   Case 2: If the GET CHALLENGE does not return SW1 SW2 == 90 00, then print

   "GET CHALLENGE was not successful. EXTERNAL AUTHENTICATE cannot be tested."

   End current test.

Test for Assertion 10.1

The APDU is tested using valid parameters.

Instantiation Scenario

1. (Pre) Construct the starting state for EXTERNAL AUTHENTICATE.
2. Issue the following EXTERNAL AUTHENTICATE APDU:

   00 82 00 00│08│cryptogram1000│ │

Verification Goal:

To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 90 00 (successful execution).
2. The currently selected object can now be updated.

Verification and Reporting Scenario:

1. **Case 1**: If the EXTERNAL AUTHENTICATE returns response code == 90 00, then verify that the selected object can now be updated:

   Issue the following UPDATE BUFFER APDU:

   80 58 00 02│05│02 _newValue5│ │

   Case 1.1: If the UPDATE BUFFER returns SW1 SW2 == 90 00 then print

   "EXTERNAL AUTHENTICATE tested using valid parameters has been verified.
   Status: Test 10.1 Passed."

Case 1.2: If the UPDATE BUFFER returns SW1 SW2 == 69 82, then print

"EXTERNAL AUTHENTICATE tested using valid parameters has not been verified because the APDU did not open the currently selected object to updating.
Status: Test 10.1 Failed."

Case 1.3: If the UPDATE BUFFER does not return SW1 SW2 == 90 00 or 69 82, then print

"EXTERNAL AUTHENTICATE tested using valid parameters has not been verified because a subsequent UPDATE BUFFER was ambiguous.
Status: Test 10.1 Undetermined.

Case 2: If the EXTERNAL AUTHENTICATE does not return response code == 90 00, then print

"EXTERNAL AUTHENTICATE tested using valid parameters has not been verified because the APDU did not return the correct response code.
Status: Test 10.1 Failed."

Test for Assertion 10.2

The APDU is tested using a bad cryptogram.

Instantiation Scenario

1. (Pre) Construct the starting state for EXTERNAL AUTHENTICATE.

2. Issue the following EXTERNAL AUTHENTICATE APDU:

```
00 82 00 00|08|_badCryptogram|  |
```

Verification Goal:
To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 69 CX (Authentication failed) or 69 83 (Authentication method blocked).

2. The currently selected object still cannot be updated.

Verification and Reporting Scenario:

1. **Case 1**: If the EXTERNAL AUTHENTICATE returns response code == 69 CX or 69 83, then verify that the selected object cannot be updated:

   Issue the following UPDATE BUFFER APDU:

   ```
   80 58 00 02|05|02 _newValue5|  |
   ```

   Case 1.1: If the UPDATE BUFFER returns SW1 SW2 == 69 82 then print

   "EXTERNAL AUTHENTICATE tested using a bad cryptogram has been verified.
   Status: Test 10.2 Passed."

Case 1.2: If the UPDATE BUFFER returns SW1 SW2 == 90 00, then print

"EXTERNAL AUTHENTICATE tested using a bad cryptogram has not been verified because a subsequent UPDATE BUFFER was successful, indicating that updating is now allowed.
Status: Test 10.2 Failed."

Case 1.3: If the UPDATE BUFFER does not return SW1 SW2 == 90 00 or 69 82, then print

"EXTERNAL AUTHENTICATE tested using a bad cryptogram has not been verified because a subsequent UPDATE BUFFER was ambiguous.
Status: Test 10.2 Undetermined.

Case 2: If the EXTERNAL AUTHENTICATE does not return response code == 69 CX or 69 83, then print

"EXTERNAL AUTHENTICATE tested using a bad cryptogram has not been verified because the APDU did not return the correct response code.
Status: Test 10.2 Failed."

Test for Assertion 10.3

The APDU is tested using a bad data field length.

This test is not part of the current version of the GSC-IS Version 2.1 Card Edge Interface Test Suite.

Test for Assertion 10.4

The APDU is tested in a context where the command is not allowed.

Instantiation Scenario

1. (Pre) Construct the starting state for EXTERNAL AUTHENTICATE.

Need to insert a "dummy" command here (the READ BUFFER) to ensure that the EXTERNAL AUTHENTICATE does not immediately follow a GET CHALLENGE.

2. (Pre) Issue the following READ BUFFER command:

   80 52 00 02|02|01 02| |.

3. Issue the following EXTERNAL AUTHENTICATE APDU:

   00 82 00 00|08|cryptogram1000| |

Verification Goal:
To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 69 85 (Conditions of use not satisfied).

2. The currently selected object still cannot be updated.

Verification and Reporting Scenario:

1. **Case 1:** If the EXTERNAL AUTHENTICATE returns response code == 69 85, then verify that the selected object cannot be updated:

Issue the following UPDATE BUFFER APDU:

```
    80 58 00 02|05|02 _newValue5|  |
```

Case 1.1: If the UPDATE BUFFER returns SW1 SW2 == 69 82 then print

"EXTERNAL AUTHENTICATE tested in a context where the command is not allowed has been verified.
Status: Test 10.4 Passed."

Case 1.2: If the UPDATE BUFFER returns SW1 SW2 == 90 00, then print

"EXTERNAL AUTHENTICATE tested in a context where the command is not allowed has not been verified because a subsequent UPDATE BUFFER was successful, indicating that updating is now allowed.
Status: Test 10.4 Failed."

Case 1.3: If the UPDATE BUFFER does not return SW1 SW2 == 90 00 or 69 82, then print

"EXTERNAL AUTHENTICATE tested in a context where the command is not allowed has not been verified because a subsequent UPDATE BUFFER was ambiguous.
Status: Test 10.4 Undetermined.

Case 2: If the EXTERNAL AUTHENTICATE does not return response code == 69 85, then print

"EXTERNAL AUTHENTICATE tested in a context where the command is not allowed has not been verified because the APDU did not return the correct response code.
Status: Test 10.4 Failed."

Test for Assertion 10.5

The APDU is tested using invalid parameters P1-P2.

Instantiatio Scenario

1. (Pre) Construct the starting state for EXTERNAL AUTHENTICATE.

2. Issue the following EXTERNAL AUTHENTICATE APDU:

```
    00 82 00 FF|08|cryptogram1000|  |
```

Verification Goal:

To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 6A 86 (Incorrect P1 or P2 parameter) or 6A 88 (Referenced data not found).

2. The currently selected object still cannot be updated.

Verification and Reporting Scenario:

1. **Case 1**: If the EXTERNAL AUTHENTICATE returns response code == 6A 86 or 6A 88, then verify that the selected object cannot be updated:

Issue the following UPDATE BUFFER APDU:

```
80 58 00 02|05|02 _newValue5|  |
```

Case 1.1: If the UPDATE BUFFER returns SW1 SW2 == 69 82 then print

"EXTERNAL AUTHENTICATE tested using a bad data field length has been verified.
Status: Test 10.5 Passed."

Case 1.2: If the UPDATE BUFFER returns SW1 SW2 == 90 00, then print

"EXTERNAL AUTHENTICATE tested using invalid parameters P1-P2 has not been verified because a subsequent UPDATE BUFFER was successful, indicating that updating is now allowed.
Status: Test 10.5 Failed."

Case 1.3: If the UPDATE BUFFER does not return SW1 SW2 == 90 00 or 69 82, then print

"EXTERNAL AUTHENTICATE tested using invalid parameters P1-P2 has not been verified because a subsequent UPDATE BUFFER was ambiguous.
Status: Test 10.5 Undetermined.

Case 2: If the EXTERNAL AUTHENTICATE does not return response code == 6A 86 or 6A 88, then print

"EXTERNAL AUTHENTICATE tested using invalid parameters P1-P2 has not been verified because the APDU did not return the correct response code.
Status: Test 10.5 Failed."

11. INTERNAL AUTHENTICATE

Starting State for Each Test:

1. A card that claims to implement the GSC-IS, Version 2.1, is in a reader. The card is subject to internal authentication.

Test for Assertion 11.1

The APDU is tested using valid parameters.

Instantiation Scenario

1. (Pre) Construct the starting state for INTERNAL AUTHENTICATE.
2. Issue the following INTERNAL AUTHENTICATE APDU:

```
00 88 00 00|08|_goodChallenge|08|
```

Verification Goal:
To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 90 00 (successful execution)
   - the data field of the response message == the correctly encrypted challenge

   or
   - SW1 SW2 == 61 08 (normal processing, data still available)
   - the correctly encrypted challenge is available to be retrieved via GET RESPONSE.

Perform this verification by inspection.

Verification and Reporting Scenario:

1. **Case 1**: If the INTERNAL AUTHENTICATE returns response code == 90 00, then manually inspect the 8 byte data field in the response message:

   Case 1.1: If the data field contains the correct 8 byte cryptogram, then print
   "INTERNAL AUTHENTICATE tested using valid parameters has been verified.
   Status: Test 11.1 Passed."

   Case 1.2: If the data field does not contain the correct 8 byte cryptogram, then print
   "INTERNAL AUTHENTICATE tested using valid parameters has not been verified by inspection.
   Status: Test 11.1 Failed."

   Case 2: If the INTERNAL AUTHENTICATE APDU returns the response code 61 08, then issue the following GET RESPONSE APDU

```
00 C0 00 00| | |08|
```

Case 2.1: If the GET RESPONSE APDU returns

- SW1 SW2 == 90 00

then manually inspect the 8 byte data field in the response message:

Case 2.2.1: If the data field contains the correct 8 byte cryptogram, then print

"INTERNAL AUTHENTICATE tested using valid parameters has been verified by inspection.
Status: Test 11.1 Passed."

Case 2.2.2: If the data field does not contains the correct 8 byte cryptogram, then print

"INTERNAL AUTHENTICATE tested using valid parameters has not been verified by inspection.
Status: Test 11.1 Failed."

Case 2.2: If the GET RESPONSE APDU does not return

- SW1 SW2 == 90 00

then print

"INTERNAL AUTHENTICATE tested using valid parameters has not been verified because a subsequent GET RESPONSE did not return the correct response code.
Status: Test 11.1 Failed."

Case 3: If the INTERNAL AUTHENTICATE does not return response code == 90 00, then print

"INTERNAL AUTHENTICATE tested using valid parameters has not been verified because the APDU did not return the correct response code.
Status: Test 11.1 Failed."

Test for Assertion 11.2

The APDU is tested using a bad data field length.

This test is not part of the current version of the GSC-IS Version 2.1 Card Edge Interface Test Suite.

Test for Assertion 11.3

The APDU is tested where the specified length of the returned challenge is incorrect.

Instantiation Scenario

1. (Pre) Construct the starting state for INTERNAL AUTHENTICATE.

2. Issue the following INTERNAL AUTHENTICATE APDU:

   00 88 00 00|08|_goodChallenge|02|

Verification Goal:
To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 6C XX (Wrong Length in L_e parameter).

Verification and Reporting Scenario:

1. **Case 1**: If the INTERNAL AUTHENTICATE returns response code == 6C 08, then print

   "INTERNAL AUTHENTICATE tested where the specified length of the returned challenge is incorrect has been verified.
   Status: Test 11.3 Passed."

   Case 2: If the INTERNAL AUTHENTICATE does not return response code == 6C 08, then print

   "INTERNAL AUTHENTICATE tested where the specified length of the returned challenge is incorrect has not been verified because the APDU did not return the correct response code.
   Status: Test 11.3 Failed."

Test for Assertion 11.4

The APDU is tested using invalid parameters P1-P2.

Instantiation Scenario

1. (Pre) Construct the starting state for INTERNAL AUTHENTICATE.

2. Issue the following INTERNAL AUTHENTICATE APDU:

   00 88 00 FE|08|_goodChallenge|08|

Verification Goal:
To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 6A 86 (Incorrect P1 or P2 parameter).

Verification and Reporting Scenario:

1. **Case 1**: If the INTERNAL AUTHENTICATE returns response code == 6A 86, then print

   "INTERNAL AUTHENTICATE tested using invalid parameters P1-P2 has been verified.
   Status: Test 11.4 Passed."

   Case 2: If the INTERNAL AUTHENTICATE does not return response code == 6A 86, then print

   "INTERNAL AUTHENTICATE tested using invalid parameters P1-P2 has not been verified because the APDU did not return the correct response code.
   Status: Test 11.4 Failed."

12. PRIVATE SIGN/DECRYPT

Starting State for Each Test:
A card that claims to implement the GSC-IS, Version 2.1, is in a reader.

Test for Assertion 12.1

The APDU is tested using valid parameters.

Instantiation Scenario

1. (Pre) Construct the starting state for PRIVATE SIGN/DECRYPT.

2. Issue the following PRIVATE SIGN/DECRYPT APDU:

   80 42 00 00│08│_goodMessageDigest│08│

Verification Goal:
To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 90 00 (successful execution)
   - the data field of the response message == the correct signature or decryption

   or
   - SW1 SW2 == 61 08 (normal processing, data still available)
   - the correct signature or decryption is available to be retrieved via GET RESPONSE.

Perform this verification by inspection.

Verification and Reporting Scenario:

1. **Case 1**: If the PRIVATE SIGN/DECRYPT returns response code == 90 00, then manually inspect the 8 byte data field in the response message:

   Case 1.1: If the data field contains the correct signature or decryption, then print
   "PRIVATE SIGN/DECRYPT tested using valid parameters has been verified.
   Status: Test 12.1 Passed."

   Case 1.2: If the data field does not contain the correct signature or decryption, then print
   "PRIVATE SIGN/DECRYPT tested using valid parameters has not been verified by inspection.
   Status: Test 12.1 Failed."

   Case 2: If the PRIVATE SIGN/DECRYPT APDU returns the response code 61 08, then issue the following GET RESPONSE APDU

   00 C0 00 00│ │ │08│

Case 2.1: If the GET RESPONSE APDU returns

- SW1 SW2 == 90 00

then manually inspect the 8 byte data field in the response message:

Case 2.2.1: If the data field contains the correct signature or decryption, then print

"PRIVATE SIGN/DECRYPT tested using valid parameters has been verified by inspection.
Status: Test 12.1 Passed."

Case 2.2.2: If the data field does not contains the correct signature or decryption, then print

"PRIVATE SIGN/DECRYPT tested using valid parameters has not been verified by inspection.
Status: Test 12.1 Failed."

Case 2.2: If the GET RESPONSE APDU does not return

- SW1 SW2 == 90 00

then print

"PRIVATE SIGN/DECRYPT tested using valid parameters has not been verified because a subsequent GET RESPONSE did not return the correct response code.
Status: Test 12.1 Failed."

Case 3: If the PRIVATE SIGN/DECRYPT does not return response code == 90 00, then print

"PRIVATE SIGN/DECRYPT tested using valid parameters has not been verified because the APDU did not return the correct response code.
Status: Test 12.1 Failed."

Test for Assertion 12.2

The APDU is tested using a bad data field length.

This test is not part of the current version of the GSC-IS Version 2.1 Card Edge Interface Test Suite.

Test for Assertion 12.3

The APDU is tested where the specified length of the returned signature or decryption is incorrect.

Instantiation Scenario

1. (Pre) Construct the starting state for PRIVATE SIGN/DECRYPT.

2. Issue the following PRIVATE SIGN/DECRYPT APDU:

```
00 88 00 00|08|_goodChallenge|02|
```

Verification Goal:
To verify the expected results:

1. The APDU returns

- SW1 SW2 == 6C 08 (Wrong Length in L_e parameter. 08 indicates the correct length).

Verification and Reporting Scenario:

1. **Case 1**: If the PRIVATE SIGN/DECRYPT returns response code == 6C 08, then print
   "PRIVATE SIGN/DECRYPT tested where the specified length of the returned signature or decryption is incorrect has been verified.
   Status: Test 12.3 Passed."

   Case 2: If the PRIVATE SIGN/DECRYPT does not return response code == 6C 08, then print
   "PRIVATE SIGN/DECRYPT tested where the specified length of the returned signature or decryption is incorrect has not been verified because the APDU did not return the correct response code.
   Status: Test 12.3 Failed."

Test for Assertion 12.4

The APDU is tested using invalid parameters P1-P2.

Instantiation Scenario

1. (Pre) Construct the starting state for PRIVATE SIGN/DECRYPT.

2. Issue the following PRIVATE SIGN/DECRYPT APDU:

   ```
   00 88 00 FE|08|_goodChallenge|08|
   ```

Verification Goal:
To verify the expected results:

1. The APDU returns
   - SW1 SW2 == 6A 86 (Incorrect P1 or P2 parameter).

Verification and Reporting Scenario:

1. **Case 1**: If the PRIVATE SIGN/DECRYPT returns response code == 6A 86, then print
   "PRIVATE SIGN/DECRYPT tested with an invalid P1 or P2 parameter has been verified.
   Status: Test 12.4 Passed."

   Case 2: If the PRIVATE SIGN/DECRYPT does not return response code == 6A 86, then print
   "PRIVATE SIGN/DECRYPT tested with an invalid P1 or P2 parameter has not been verified because the APDU did not return the correct response code.
   Status: Test 12.4 Failed."

Appendix A

List of Symbolic Constants

```
_ACR1ID == 01

_ACR2ID == 02

_applet1AID == A0000000791111 (CAC), or
               A0000001161111 (GSC)

_applet1AIDLen == 07

_applet2AID == A0000000792222 (CAC), or
               A0000001162222 (GSC)

_applet2AIDLen == 07

_applet3AID == A0000000793333 (CAC), or
               A0000001163333 (GSC)

_applet3AIDLen == 07

_applet4AID == A0000000794444 (CAC), or
               A0000001164444 (GSC)

_applet4AIDLen == 07

_applet5AID == A0000000795555 (CAC), or
               A0000001165555 (GSC)

_applet5AIDLen == 07

_appletPINAID == A0000000796666 (CAC), or
                 A0000001166666 (GSC)

_appletPINID == 07

_appletXAUTHAID == A0000000797777 (CAC), or
                   A0000001167777 (GSC)

_appletXAUTHID == 08
_badAppletAID == AAAAAAAAAAAAAA

_badAppletAIDLen == 07

_badCryptogram == 00 00 00 00 00 00 00 00

_badCryptogramLen == 08

_badObjectID == FE

_badPIN ==  00 00 00 00 00 00 00 00
_badPINLen == 08
```

```
_existingValue1 == 01 01 01 01

_existingValue2 == 02 02 02 02

_existingValue3 == 03 03 03 03

_existingValue4 == 04 04 04 04

_existingValue5 == 05 05 05 05

_existingValue6 == 06 06 06 06

_existingValue7 == 07 07 07 07

_existingValue8 == 08 08 08 08

_existingValue9 == 09 09 09 09

_existingValue10 == 10 10 10 10

_existingValue11 == 11 11 11 11

_goodChallenge == 01 02 03 04 05 06 07 08

_goodPIN == 01 02 03 04 05 06 07 08

_goodMessageDigest ==  08 07 06 05 04 03 02 01
_newTag == 11

_newValue1 == 11 11 11 11

_newValue2 == 12 12 12 12

_newValue3 == 13 13 13 13

_newValue4 == 14 14 14 14

_newValue5 == 15 15 15 15

_object11ID == 00 01

_object21ID == 00 02

_object31ID == 00 03

_object41ID == 00 04

_object51ID == 00 05

_tag1 == 01

_tag2 == 02

_tag3 == 03

_tag4 == 04
```

```
_tag5 == 05

_tag6 == 06

_tag7 == 07

_tag8 == 08

_tag9 == 09

_tag10 == 0A

_tag11 == 0B
```

Appendix B

Description of the Card Required for Testing VCEI (VM Card) Implementations

- There exist at least seven applets on the card:

1. An applet with the following properties:
    - AID == _applet1AID. This AID is of length == _applet1IDLen
    - manages precisely one object, whose object ID == _object11ID. This object
        - has T-buffer: 00 06 _tag1 04 _tag2 08
        - has V-buffer: 00 0E _existingValue1 _existingValue2
        - has access control rule ACR_ALWAYS for all services
    - does not contain an object with object ID == _badObjectID.

2. An applet with the following properties:
    - AID == _applet2AID. This AID is of length == _applet2AIDLen
    - manages precisely one object, whose object ID == _object21ID. This object
        - has T-buffer: 00 04 _tag8 04
        - has V-buffer: 00 06 _existingValue8
        - has access control rule ACR_XAUTH_AND_PIN for all services.

3. An applet with the following properties:
    - AID == _applet3AID. This AID is of length == _applet3AID
    - manages precisely one object, whose objectID == _object31ID. This object
        - has T-buffer: 00 04 _tag9 04
        - has V-buffer 00 06 _existingValue9
        - has access control rule ACR_PIN for all services. _goodPIN is the PIN that has been set for each service of the object, and 3 is the maximum number of PIN tries allowed.

4. An applet with the following properties:
    - AID == _applet4AID. This AID is of length == _applet4AID
    - manages precisely one object, whose objectID == _object41ID. This object
        - has T-buffer: 00 06 _tag10 04 _tag11 04
        - has V-buffer: 00 0A _existingValue10 _existingValue11
        - has access control rule ACR_ALWAYS for each service.

5. An applet with the following properties:
    - AID == _applet5AID. This AID is of length == _applet5AID
    - manages precisely one object, whose objectID == _object51ID. This object
        - has T-buffer: 00 0C _tag3 04 _tag4 04 _tag5 04 _tag6 04 _tag7 04
        - has V-buffer: 00 16 _existingValue3 _existingValue4 _existingValue5 _existingValue6 _existingValue7

  - has access control rule ACR_ALWAYS for its read service, and ACR_XAUTH for its update service.

6. An applet with the following properties:
   - AID == _appletXAUTHAID
   - applet ID == _appletXAUTHID
   - keyIDOrReference == 01
   - manages the access control method XAUTH.

7. An applet with the following properties:
   - AID == _appletPINAID
   - applet ID == _appletPINID
   - keyIDOrReference == 01
   - manages the access control method PIN.

- The card does not have an applet whose AID == _badappletAID, of length == _badAppletAIDLen.

- The card's CCC has at least two entries in its ACR table:

1. ACRID == ACR1ID, corresponding to ACR_PIN.

2. ACRID == ACR2ID, corresponding to ACR_XAUTH_AND_PIN.